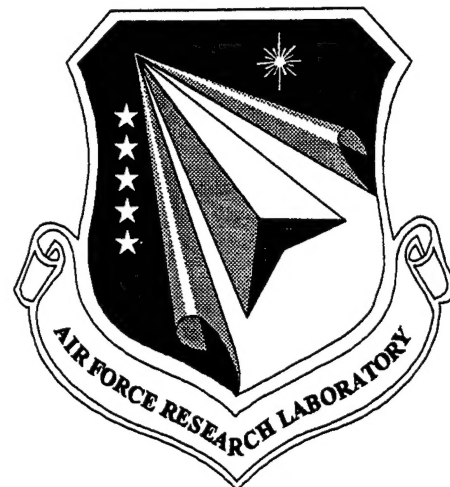AFRL-PR-WP-TR-1999-2069

# ROBUST, MULTIVARIABLE, QUANTITATIVE DESIGN OF AN ADAPTIVE MODEL-BASED CONTROL FOR JET ENGINES

ZANE D. GASTINEAU

AFRL/PRTA
1950 FIFTH ST
WRIGHT-PATTERSON AFB, OH 45433

JUNE 1998

FINAL REPORT FOR 08/01/95 – 06/07/98

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

**19991005 006**

PROPULSION DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-7251

ZANE D. GASTINEAU
Controls and Diagnostics
Engine Integration and Assessment Branch
Turbine Engine Division
Propulsion Directorate

RICHARD J. KRABAL
Chief
Engine Integration and Assessment Branch
Turbine Engine Division
Propulsion Directorate

JOHN T. DATKO
Acting Chief of Technology
Turbine Engine Division
Propulsion Directorate

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 |
|---|---|---|---|

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE JUNE 1999 | 3. REPORT TYPE AND DATES COVERED FINAL REPORT FOR 08/01/1995 - 06/07/1998 |
|---|---|---|

**4. TITLE AND SUBTITLE**
ROBUST, MULTIVARIABLE, QUANTITATIVE DESIGN OF AN ADAPTIVE MODEL-BASED CONTROL FOR JET ENGINES

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

ZANE D. GASTINEAU

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
AFRL/PRTA
CONTROLS AND DIAGNOSTICS
ENGINE INTEGRATION AND ASSESSMENT BRANCH
TURBINE ENGINE DIVISION
PROPULSION DIRECTORATE

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
PROPULSION DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB, OH 45433-7251
POC: ZANE D. GASTINEAU, AFRL/PRTA, 937-255-6690

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

AFRL-PR-WP-TR-1999-2069

**11. SUPPLEMENTARY NOTES**

THIS REPORT IS A DISSERTATION

**12a. DISTRIBUTION AVAILABILITY STATEMENT**

APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

The robust design of a model-based control system for a single-spool gas turbine engine is presented. The overall objective of this work is to develop a robust, quantitative design methodology that allows the turbine engine control system engineer the ability to control variables that can not be directly measured. The study makes use of the model-based control structure that is of current interest within the turbine engine control community. Then the concept of robustness is added to the model-based control structure to ensure satisfactory operation of the turbine engine throughout the flight envelope.

**14. SUBJECT TERMS**
Model-based control, tracking Filter, Adaptive Control, Robust control, Quantitative Feedback Theory

**15. NUMBER OF PAGES**
156

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | SAR |

Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

# TABLE OF CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

Gastineau, Zane D.
                    B.S., University of Kentucky, 1987
M.S., Wright State University, 1993

Robust, Multivariable, Quantitative Design of an
Adaptive Model-Based Control for Jet Engines

Advisor: Professor Osita D.I. Nwokah
Co-Advisor: Dr. Gemunu Happawana

Doctor of Philosophy degree conferred: August 7, 1998

Dissertation completed: June 7, 1998

In this dissertation, the robust design of a model-based control system for a single-spool gas turbine engine is presented. The overall object of this work is to develop a robust quantitative design methodology that allows the control system engineer the ability to control variables that can not be directly measured. The study makes use of the model-based control structure that is of current interest within the turbine engine control community. Then the concept of robustness is added to the model-based control structure to ensure satisfactory operation of the turbine engine throughout the flight envelope.

In Chapter 1, a detailed explanation of the objectives and significance of this work is given along with a description of the previous research work done in this area. A brief description of nonlinear engine modeling and the generation of linear models is given in Chapter 2. In addition, Chapter 2 contains a methodology for the selection of appropriate models for the design process. The theoretical development for the design of a tracking filter is presented in Chapter 3. Chapter 4 shows the theoretical development of a robust control design technique that provides both decoupling and robustness. Chapter 5 presents a detailed design example of the entire design process. In Chapter 6 conclusions are drawn and suggestions for future work are offered.

# ACKNOWLEDGEMENTS

To God I owe all, without Him none of this would have been possible.

My deepest appreciation goes to my adviser, Dr. Osita D. I. Nwokah, for his help in my graduate study, for his guidance and encouragement, and for providing the opportunity to achieve this work. My sincerest appreciation to my co-advisor and friend, Dr. Gemunu Happawana, for his friendship and the endless time spent discussing this work and many other subjects. I would also like to thank the members of my committee Drs. David Johnson, Yildirim Hurmuzlu and Alireza Khotanzad for their careful reading and comments on this manuscript. I am very grateful to the faculty, staff and students of the Department of Mechanical Engineering for their help and support.

I thank the Air Force Research Laboratory, Propulsion Directorate, for the time away from my normal duties to pursue this work. I also thank, Mr. Charles Skira, who initially encouraged my pursuit of this work and Mr. Timothy Lewis who took on my duties in addition to his own while I was gone.

I am very grateful for my parents, Ron and Sue Gastineau, for their undying love, endless support and unwavering confidence in my abilities during the pursuit of my dreams. Also, special thanks to my wife's mother, Madine Lawson, for her encouragement and love.

Finally, I owe the deepest thanks to my beloved wife, Carole, for love, encouragement and patience. Without her support this work would not be possible.

# DEDICATION

This dissertation is dedicated to my children, Isaac and Caitlyn Gastineau, who are at the beginning of their educational journey. Always remember that with God, hard work and perseverance, you can achieve your dreams.

*"...if God is for you, who can be against you." Romans 8:31*

ROBUST, MULTIVARIABLE, QUANTITATIVE DESIGN OF AN

ADAPTIVE MODEL-BASED CONTROL FOR JET ENGINES

A Dissertation Presented to the Graduate Faculty of the

School of Engineering and Applied Science

Southern Methodist University

In

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Mechanical Engineering

by

Zane D. Gastineau

(B.S., University of Kentucky, 1987)
(M.S., Wright State University, 1993)

August 7, 1998

# CHAPTER 1

## INTRODUCTION

Military aircraft of the future are envisaged to be multirole vehicles. This dictates the requirement for multimode integration of airframe and propulsion systems over a wide flight envelope. In most present designs, the multimode requirement is being met by varying aircraft configurations or relying on active controls technology to accommodate response changes due to varying flight conditions. It is becoming evident that in order to meet this variable capability much of it must be met by the propulsion system.

Advanced aircraft turbine engines have been the focus of extensive research in recent years under the Integrated High Performance Turbine Engine Technology (IHPTET) program. The IHPTET program is a national initiative sponsored by the Department of Defense (DoD) and the National Aeronautics and Space Administration (NASA) seeking to improve the overall performance, reliability and affordability of advanced gas turbine engines (Brown, 1990). This manifests itself in the need for smaller engines that produce the same or higher levels of thrust while having better fuel economy and costing less to maintain.

IHPTET is a multiphase program with very specific goals to be accomplished during each phase. In order to achieve these goals the turbine engine was broken down into its major component areas and objectives were established for each component that would then be related to or influence the overall goals of the program (Petty and Henderson, 1987). For the propulsion control system these objectives are weight reduction and design margin reduction. The weight reduction goal has been addressed by material-for-material substitution in the major control components such as fuel pumps and actuators. The design margin goal looks more at the overall impact that improvements in the control system architecture and logic can have on the engine system. This second goal is the one being addressed by the work described herein.

## 1.1    Review of Relevant Research

The application of multivariable time- and frequency-domain control design techniques to aircraft turbine engine control problems has been widely published in the literature [see (Merrill, *et al.*, 1984) for an extensive list of references]. Many of the published solutions, however, suffer from excessive controller order and complexity, necessitating the use of complex control logic to implement the control law. The development of multivariable frequency-domain control design techniques such as Rosenbrock's (1969) Inverse Nyquist Array (INA) promised the use of relatively inexpensive analog hardware or simple digital control software to implement the control law. This has not proven to be the case in that many of the published solutions are only valid at a single operating point in the flight envelope, necessitating the use of digital

2

hardware and gain schedules to implement a different control law for each point in the flight envelope (Merrill, *et al.*, 1984; Polley, *et al.*, 1988). This has the undesirable side effect of requiring extensive test cell and installed engine-controller testing, which is very expensive. The applicability of frequency-domain control design techniques to this type of nonlinear control problem, however, has been demonstrated (Leininger, 1981).

The development of Quantitative Feedback Theory (QFT) to handle uncertain multivariable control problems has enjoyed tremendous success over the past decade, particularly in the area of controller design for plants that may be accurately modeled by connected sets of parametrically uncertain differential equations (Horowitz, 1991). Indeed, QFT based techniques have been applied to turbofan engine control problems in the past, such as the work of Yau et al. (1994), however, an appeal to plant high gain stabilizability was resorted to. Due to the inherent time delay behavior of turbomechanical systems, this is not a very realistic assumption. Recently, the Inverse Nyquist Array design technique (Rosenbrock, 1970) was modified by Nwokah and Nordgren (1993) using techniques from QFT and H∞ control theory (Zames, 1981) to handle multivariable control problems for uncertain plant sets with both parametric and unstructured uncertainty. This decentralized design technique, termed the Quantitative Nyquist Array (QNA), offers considerable advantage over design techniques that treat model uncertainty in a single manner.

The technique of μ-synthesis is rooted in H∞ control theory (Zames, 1981) and has been pursued extensively by Owen and Zames (1981); Doyle, et al., (1992); and Chaing and Safanov, (1988). μ-synthesis seeks to eliminate some of the performance

3

weaknesses of the standard H∞ theory such that desired performance measures are achieved for all plant set elements. This theory has been expanded to handle plants with both parametric and unstructured uncertainty models. Quantitative Feedback Theory and μ-synthesis offer, perhaps, the greatest promise in the development of a single multivariable robust control law that is capable of delivering desired engine performance levels throughout the flight envelope.

Multimodal engine control was first looked at under the Air Force funded Intelligent Engine Control (IEC) (Adibhatla, *et al.*, 1992) program and was subsequently followed by studies conducted by NASA-Lewis Research Center. From these studies it is obvious that conventional control techniques would not satisfy all the needs for the future. Therefore, a new paradigm was needed to meet future requirements. This is when the technique of model-based control was introduced by Adibhatla, *et al.* (1992) and Qi and Maccallum (1993).

## 1.2    Significance of the Work

The main objective of turbine engine control systems is to adjust the system for rapidly required thrust changes, while maintaining stable operation of the compressor and keeping the turbine inlet temperature within safety limits. Usually specific engine performance rating points are generally defined as steady-state design goals. In addition to regulating the engine for steady-state operation the control must constrain the operation within various aerodynamic, thermodynamic, and mechanical or structural design limits. Typical aerodynamic limits include fan and compressor stability and maximum allowable

4

engine inlet airflow distortion. Thermodynamic limits include minimum combustor or augmentor fuel/air ratios for stable combustion, minimum fuel flow for ignition and maximum fuel/air ratios. Mechanical or structural limits consist of maximum allowable rotor speeds, maximum allowable nozzle flap temperatures, maximum combustor case pressure, rotor creep limits, and maximum average and peak values of turbine blade metal temperature. Steady-state performance rating points at sea level static and altitude along with transient response characteristics are specified in "General Specification for Turbojet and Turbofan Engines" (MIL-E-5007D).

The first turbine engines used hydromechanical control systems to meet all the requirements levied on the propulsion system. However, the need for increased functionality, reliability and maintainability led to the introduction of digital engine controls in the early 1980's. According to Skira and Agnello (1991) today's turbine engine control systems are greatly improved over the first electronic control systems. However, these first digital electronic controls did little more than emulate the hydromechanical control they replaced. Since that time much work has been done on improving these systems by adding redundancy and evolving the components that go into them with little change in the control strategy. The primary emphasis has been on weight reduction with very little attention given to the power provided by today's advanced computers.

Protection of the engine hardware is one of the primary responsibilities of the control system. This consists of avoiding all the limits listed above. In order to avoid reaching these limits, margins (safety factors) are established that consist of a worst case

5

stackup of undesirable effects that could cause the engine to exceed one or more of the limits. In the case of stall margin, this consists of effects due to but is not limited to values associated with: inlet distortion, engine transients, Reynolds number effects, engine-to-engine variations and control tolerances. Historically, the amount set aside for these design margins has not changed.

The control system can not create performance that the machinery does not already have built into it. However, performance is designed out of a system in order to build in large safety factors. Historically these margins are constant even though there has been significant technological advancements. This margin stackup is a series of worst case events that could occur for each effect. It rarely occurs that the maximum effect in each parameter occurs at the same time. Therefore, if the margin stackup is used in a smart way we can regain some of the lost performance while protecting the engine and not sacrificing the safety or integrity of the machinery.

In order to illustrate these concepts consider Figure 1.1. In this plot, thrust ratio is plotted as function of temperature ratio. Temperature ratio is defined as the turbine inlet temperature limit divided by the turbine inlet temperature at stoichiometric temperatures. This axis then reflects in general the turbine engine material limitations that exist over time. Thrust ratio is then defined as the thrust produced at a given stall margin and temperature ratio divided by the thrust at no stall margin and stoichiometric temperatures. The line labeled stall line is the line of zero margin and the one labeled operating line is a line of constant design stall margin. The area between these two curves is the performance that is lost due to constant design stall margins. The ellipses represent

6

engine generations or level of engine technology as time progresses. The wedge labeled recaptured performance is shown this way to illustrate that understanding of the effects of the parameters that make up the margin stackup are not fully understood now. As we gain further understanding the amount of performance that can be regained will increase.
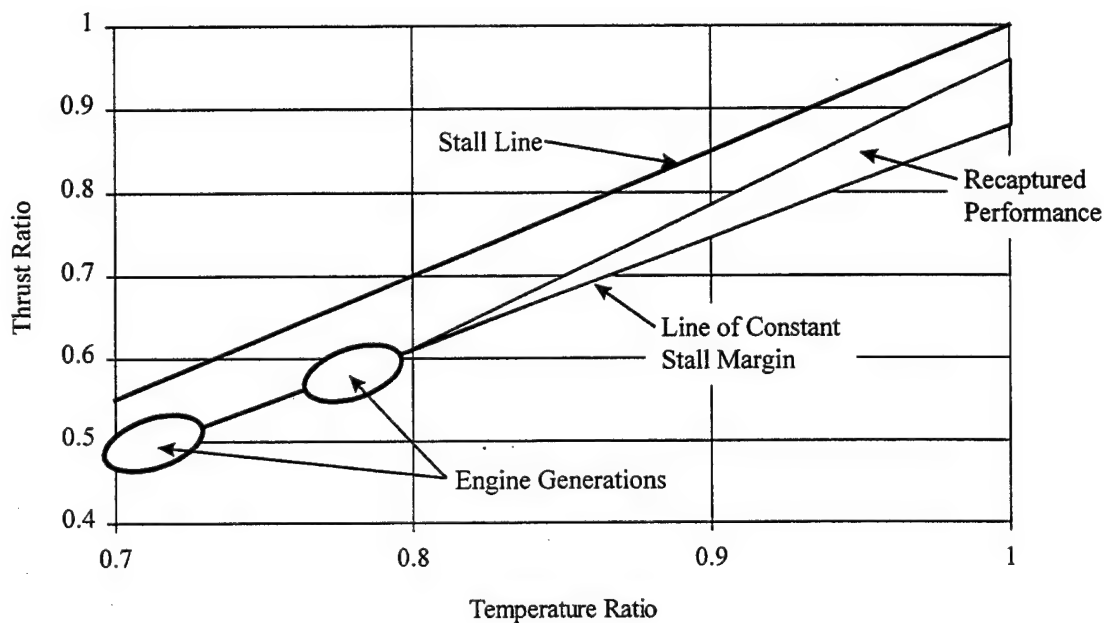


**Figure 1.1 Recaptured Performance Potential**

In order to recapture this loss performance a better understanding of the processes that influences each factor in the margin stackup is necessary and a means of determining the conditions where each factor is dominant needs to be found. Once this has been accomplished then the control can offer additional performance by using up the margin or providing engine protection when the operating conditions require it.

## 1.3 Conventional Versus Model-Based Control

During any engine development process a great deal of time is spent creating a detailed nonlinear simulation of the engine. The control engineer usually starts his work after the simulation is created. First the nonlinear model is linearized. Then the linear model responses are checked against the nonlinear model responses. Finally, the control design proceeds with the linear models. It is at this point in the design process that significant improvement can be made if the problem is rethought slightly.

This nonlinear model that is developed contains all the knowledge and expertise available for that engine at that time. But, it is never used for anything more than evaluation of proposed changes to control logic or engine hardware. The question is, "Is there a way that the knowledge contained within the nonlinear simulation can be used to improve the functionality of the engine?" The answer is yes, through the control methodology termed model-based control. Now consider the differences between conventional and model-based control.

The conventional control system is shown in Figure 1.2. In this type of architecture, sensed parameters such as fan speed and pressure ratio are used as feedback signals to generate error signals that the control uses to drive actuators to their correct position. These sensed parameters are used because they infer what the parameters of interest (thrust and stall margin) should be. Because this inference is never exact, large margins (safety factors) are established in order to protect the engine.

8
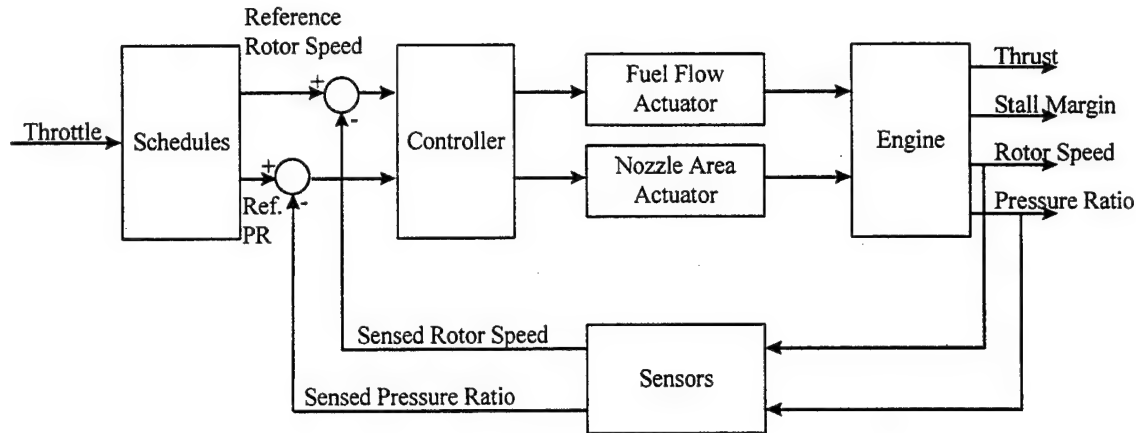
**Figure 1.2 Conventional Control System**

The model-based control architecture is shown in Figure 1.3. The key to this architecture is that a model (a computer program representation of the plant) is included in the feedback path. This model can compute variables of interest, which can not be measured or are not easily measured. These computed values are then used to generate error signals that the controller uses to drive the actuators to their desired positions. The main advantage of having a resident model is that the control can be done on variables of interest (i.e., thrust and stall margin). Then the margins can be managed more efficiently thereby allowing for an increase in performance.

The key to model-based control is the tracking filter. Its sole purpose is to guarantee that the model is matching the plant. This becomes somewhat of a problem when there are only a limited number of sensed parameters available to match. In addition, the parameters to update can become enormous. In general, the tracking filter is square, the same number of inputs and outputs. However, current research is directed

9

toward the development of methods for generating nonsquare tracking filters. A working nonsquare tracking filter has not yet been found. The tracking filter is expected to work well in whatever mode the engine is operating in. This also presents a problem in that the updates tend to distort the model in such a way as to optimize the parameter of interest. This parameter does not have to be the same for every mode of operation. Then a tracking filter that works well for every mode of operation, termed a multipurpose tracking filter, is needed. However, to date no such filter exists. There are solutions to this problem such as multiple copies of the model each with their own tracking filter for each mode of operation. But these solutions are not very viable for aircraft engine control applications due to the lack of physical space to put additional memory in for additional models and tracking filters.
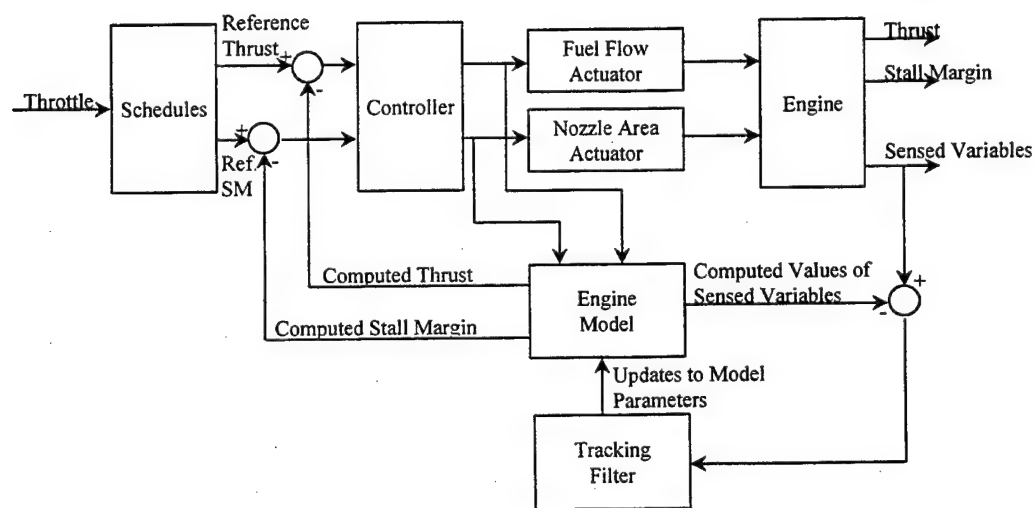


**Figure 1.3 Model-Based Control System**

10

## 1.4 Other Control Methodologies that Use Internal Models

There are two main control design techniques that exist in today's literature that make use of a model in the control: Model-Reference Adaptive Control (MRAC) and Internal-Model Control (IMC). There are key differences between Model-Based Control (MBC) described in this work and these existing techniques. However, one of the main differences in every case is how the error signal, generated by comparing the outputs of the plant and model, is used.

Model-Reference Adaptive Control (MRAC) is one of the main approaches to adaptive control. The MRAC system shown in Figure 1.4 consists of a plant (P), a reference model (M), a feedback regulator (controller) (C) and an adaptation system. The reference model contains the desired performance and will express the desired output to the external command. The feedback regulator has adjustable parameters that are varied through the adaptation system based on the error between the reference model and the plant. The plant outputs are also used for traditional feedback to compare with the reference input to generate a signal for the feedback regulator. The MRAC approach was developed to handle the case where the specifications are given in terms of a desired response model (Astrom and Wittenmark, 1989).

Internal-Model Control (IMC) was initially applied to solve problems in the process control industry, and is related to disturbance accomodation in the system, where empirical or dynamical models of the processes under control are available (Morari and Zafiriou, 1989). The IMC system shown in Figure 1.5 consists of a plant (P), a plant

11

model (PM) and a feedback regulator (C). In this approach, the error signal generated by comparing the plant and plant model outputs is used as the feedback signal.
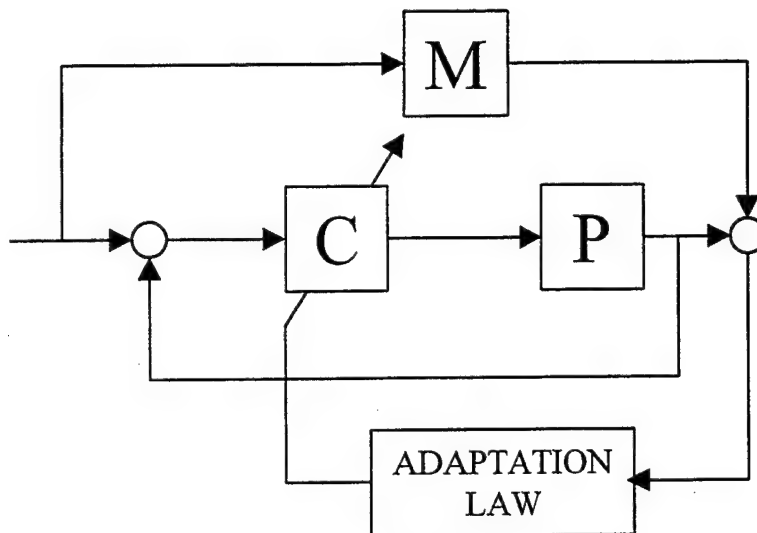


**Figure 1.4 Model-Reference Adaptive Control Structure**
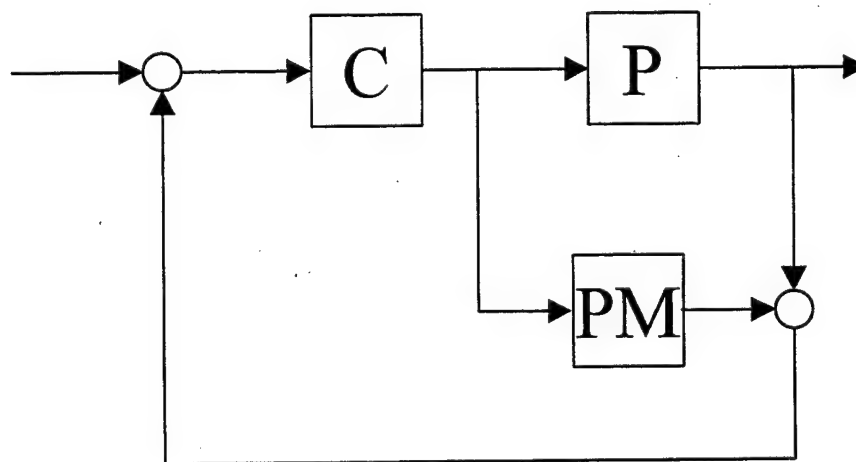


**Figure 1.5 Internal-Model Control Structure**

The Model-Based Control (MBC) approach described in Section 1.3, which is different from MRAC and IMC, was developed to solve the problem of controlling plant outputs that are not directly measurable. The key to this architecture is that the plant model variables are changing due to the adaptation error signal (the difference between the plant an the plant model outputs, when subjected to the same input) fed into the adaptation system (the tracking filter described in Chapter 3). When this error signal is below a certain threshold the plant model computes the desired feedback variables that are compared to the standard reference inputs, to generate the reference error signal which is fed to the feedback regulator for tracking response.

## 1.5    Overview

In this work a concise and workable methodology for the design of a robust multivariable model-based control system will be presented. The key areas to be addressed are the design of the tracking filter, and the design of a robust control law using unmeasurable feedback variables. While model-based control techniques have been discussed, they have never been pursued in the context of robustness.

The remainder of this document is arranged in the following manner. A complete description of the nonlinear engine model developed for this effort and its linearization is given in Chapter 2. The tracking filter is considered in Chapter 3. In Chapter 4 the robust multivariable control design problem is discussed along with H-matrix theory. A detailed design problem is presented in Chapter 5. Chapter 6 summarizes the contributions of this research, and discusses some future research topics. Appendix A

contains a well-known methodology for finding the transition matrix required to convert an observable system into the observable canonical form. Appendix B contains proofs of important theorems given throughout this dissertation. The Computer routines used to perform the work presented herein are given in Appendix C. Finally, in Appendix D the normalized linear models are given.

# CHAPTER 2

# TURBINE ENGINE FUNDAMENTALS AND MODELING

## 2.1 Introduction

The propulsion system of choice for a wide variety of land, sea, and air vehicles is the turbine engine. Because of varied operational and performance requirements, future aircraft systems will impose unprecedented challenges upon the system designer. Advanced military aircraft contemplated for the future and beyond will likely be designed as multimission aircraft with both air-to-air and air-to-ground capabilities. These same systems will possess many of the following operational requirements: Short Take-Off and Vertical Landing (STOVL) capabilities, automatic threat evasion, terrain avoidance/threat avoidance, and automatic weapon delivery. In addition, reliability and maintainability requirements will be much higher for future systems (Adibhatla, *et al.,* 1992). Therefore, an understanding of the basic principles of turbine engine design and modeling is paramount to understand the work presented herein.

## 2.2 Engine Fundamentals

The primary purpose of the turbine engine (shown in Figure 2.1) is to impart a change in momentum to a mass of fluid. This change in momentum is equivalent to an acceleration of a working fluid producing an external force (Thrust) on the system. In a

turbojet this is accomplished by bringing air into the inlet, compressing the mass of air, mixing fuel with the high pressure air in the combustor section there igniting and burning it, the fluid is expanded in the turbine section driving the turbines which in turn power the compressors. The fluid is further expanded through the nozzle section to a high velocity (conversion of pressure and thermal energy into kinetic energy) thus increasing the momentum of the fluid and producing thrust.
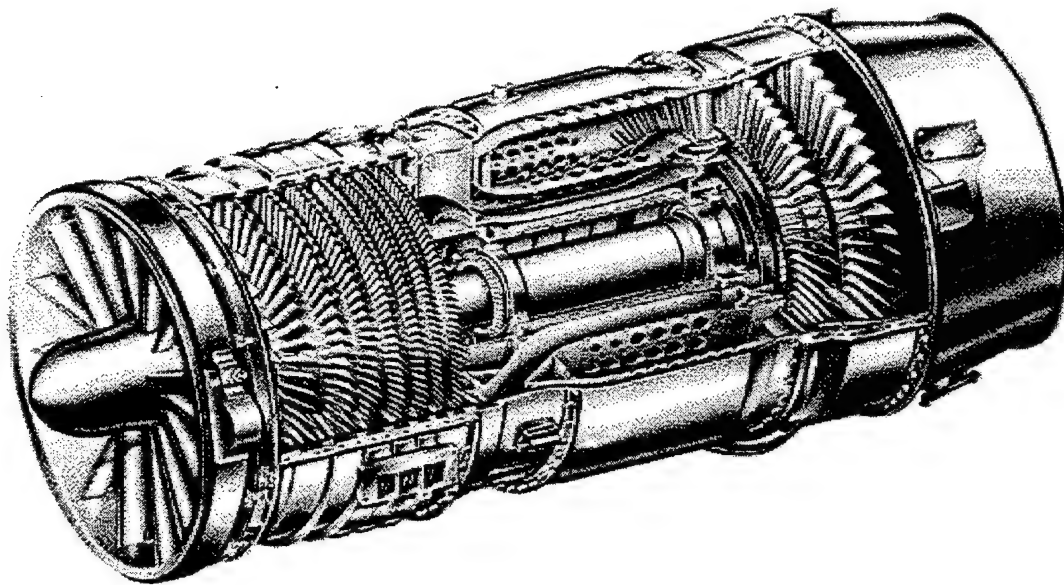


**Figure 2.1 J85 Nonafterburning Engine**

In addition to acting as the propellant fluid, the air acts as the working fluid in a thermodynamic process. This process can be represented thermodynamically by the Air Standard Brayton Cycle as shown in Figure 2.2 for the turbojet. The processes that make up this cycle are given in Table 2.1.

16
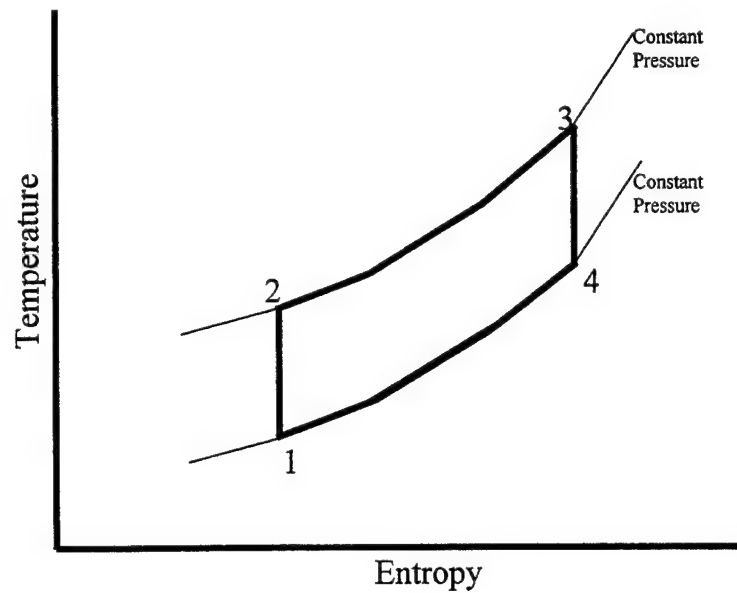
**Figure 2.2 The Brayton Cycle**

**Table 2.1 Brayton Cycle Processes**

| Leg | process description |
|-----|---------------------|
| 1 – 2 | Reversible, adiabatic (isentropic) compression between minimum and maximum pressures |
| 2 - 3 | Heat addition at constant maximum pressure |
| 3 – 4 | Reversible, adiabatic (isentropic) expansion between maximum and minimum pressures |
| 4 - 1 | Heat rejection at constant minimum pressure |

17

The behavior of turbine components is normally represented on "maps". The performance maps for compressor and turbines normally consist of the following parameters: component pressure ratio, corrected mass flow rate, a corrected speed parameter, and adiabatic efficiency. A typical compressor map is shown in Figure 2.3. The curve marked stall line represents a limit on compressor performance. Operation below this line is essential for satisfactory engine performance. Steady operation above this line is impossible and entering this region even momentarily is hazardous to the engine structural integrity.
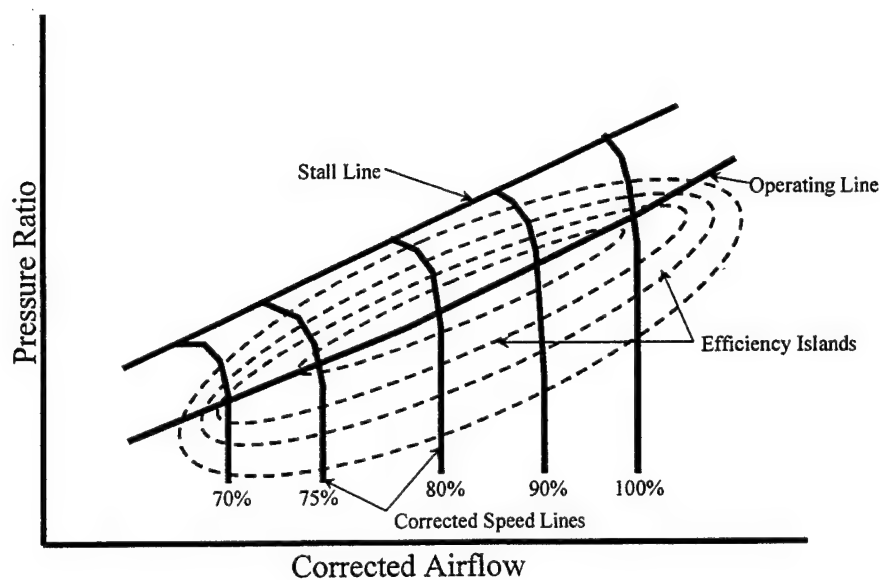


**Figure 2.3 Compressor Map**

An example turbine map is shown in Figure 2.4. The main difference is that on this map it is the reciprocal pressure ratio plotted as a function of corrected mass flow rate and corrected speed. Since the turbine is usually choked, other methods of modeling the turbine can be used just as effectively if a map is not available. One such method is using the continuity equation and considering the turbine pressure distribution as a series of pressure drops across the elements. This is the method used for this work.
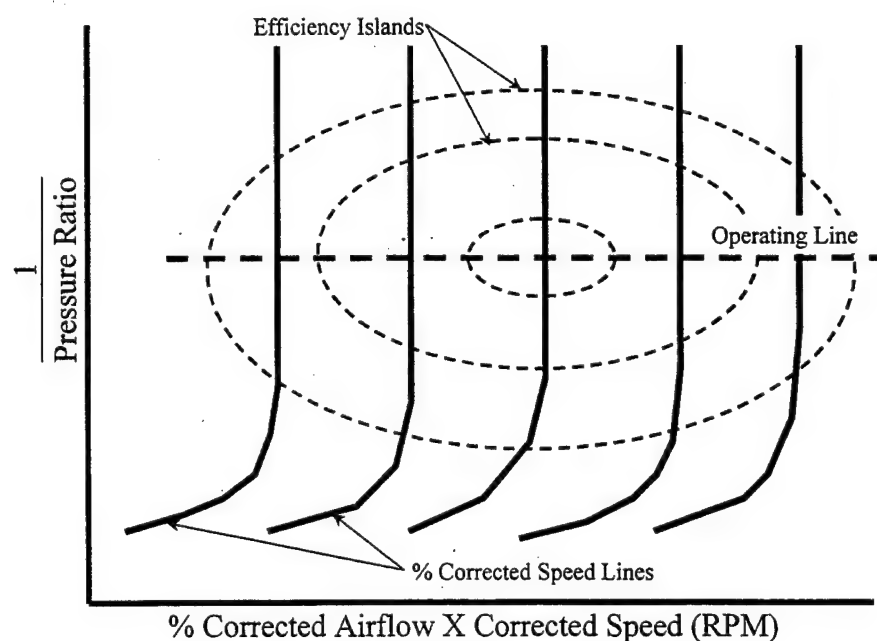


**Figure 2.4 Turbine Map**

The engine community has developed a systematic notation to facilitate the manipulation of equations and to describe at what point along the gas path that is being discussed. The work herein will follow the standardized gas turbine engine station

19

identification and nomenclature system recommended by SAE, Inc., Aerospace Recommended Practice (ARP 755A). Each station location refers to a particular engine location. These locations are identified in Table 2.2. See Figure 2.5 for a schematic representation.

**Table 2.2 Station Designation and Associated Locations**

| Station | Location |
|---------|----------|
| 0 | Far Upstream |
| 1 | Inlet Entry |
| 2 | Compressor Entry |
| 3 | Compressor Exit |
| 4 | Combustor Discharge |
| 5 | Last Turbine Discharge |
| 6 | Available for Mixer, Afterburner, etc. |
| 7 | Engine/Exhaust Nozzle Interface |
| 8 | Exhaust Nozzle Throat |
| 9 | Exhaust Nozzle Discharge |
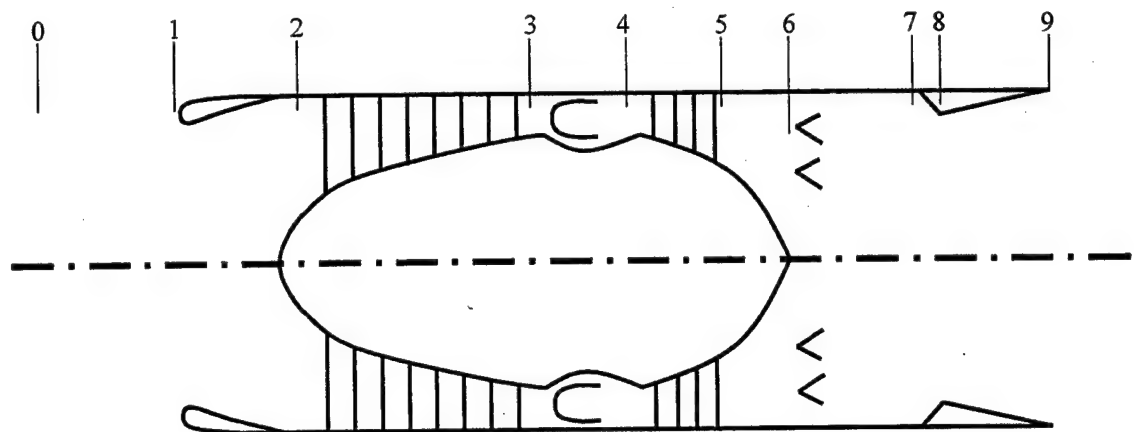
**Figure 2.5 Turbojet with Station Number Locations**

## 2.3 Engine Modeling

A simulation or mathematical representation of gas turbine engines is fairly common and has been reported by several authors: Szuch and Seldner (1975), French (1982), Mattingly, *et. al.* (1987), Oates (1988), Seldner, *et. al.* (1971), Sellers and Terren (1974) and Szuch (1974). For this work a simulation of a single-spool turbojet engine was developed using thermodynamic, aerodynamic, mechanical, or empirical relationships of each of the major components. The model was adjusted to match actual test data of a turbojet. This matching was accomplished through efficiency adjustments.

Maps were used to establish the airflow and thermodynamic performance of the compressor. The map was changed into lookup tables of temperature ratio and corrected airflow for conditions of corrected speed and pressure ratio. Performance of other components was established to agree with the compressor data and engine test data. Due to the proprietary nature of the maps they can not be printed in this document.

21

**2.3.1 Compressor Efficiency.** The compressor efficiency at partial power was modified to a lower value in order to produce a fuel flow that is more representative of the value obtained during actual engine testing. The efficiency was modified so that, for the idle speed used in the simulation the efficiency would be 75 percent as opposed to the 84 percent obtained from the map. The efficiency matched well at military speed. Therefore, the efficiency was blended from idle (68% corrected speed) to military speed (100% corrected speed) where no correction was needed. The efficiency correction is accommodated in the simulation by dividing the temperature rise, $T_3$ - $T_2$, computed from the map by the correction factor.

**2.3.2 Compressor Torque.** The compressor torque was obtained by equating the heat energy per unit time of the compressor gas to the work per unit time required to drive the compressor. The work per unit time defined by

$$\frac{\text{ft.-lb.}}{\text{minute}} = (\tau_c)(N)2\pi \qquad [2.1]$$

The power required based on the heat consideration is defined as

$$h = c_{p_c}\dot{m}_c(T_3 - T_2) \qquad [2.2]$$

Equating equations [2.1] and [2.2] in the proper units and solving for the torque, results in the following relationship.

$$\tau_c = \frac{K\dot{m}_c c_{p_c}(T_3 - T_2)}{N} \qquad [2.3]$$

22

In addition, equation [2.3] must be modified by the efficiency correction mentioned above by dividing the temperature rise by the correction.

**2.3.3 Combustor Equations.** The combustor performance is defined by the balance of the energy added by the fuel equated to the energy required to produce the temperature rise in the combustor. The energy produced by the fuel is considered to be,

$$h_f = \dot{m}_f(Q)\eta_b \qquad [2.4]$$

The energy required to produce the $(T_4 - T_3)$ temperature rise is defined as,

$$h_b = c_{p_b}\dot{m}_4(T_4 - T_3) \qquad [2.5]$$

Equating equations [2.4] and [2.5] then solving for temperature rise results in the following formula,

$$(T_4 - T_3) = \frac{Q\eta_b\dot{m}_f}{c_{p_b}\dot{m}_4} \qquad [2.6]$$

In order to increase the steady-state idle fuel flow in the simulation to make it agree with the engine test data, the efficiency of the combustor is varied with speed similar to the method used for compressor efficiency.

23

**2.3.4 Turbine Gas Flow.** Ideally a turbine map would have represented the turbine; however, none was available. The flow equation was established from known information. The continuity equation was the basic equation used. The turbine pressure distribution is considered to be the result of a series of pressure drops that represents the two rotor and two nozzle diaphragms. A split of 45 percent, 55 percent was considered for the two stages. The pressure distribution across each stage was considered to be 60 percent for the diaphragm to 40 percent for the rotor. Using this relationship, the drop across the assembly was represented as:

| | |
|---|---|
| 1st diaphragm | $\left(P_5/P_4\right)^{(0.45)(0.60)} = \left(P_5/P_4\right)^{0.27}$ |
| 1st rotor | $\left(P_5/P_4\right)^{(0.45)(0.40)} = \left(P_5/P_4\right)^{0.18}$ |
| 2nd diaphragm | $\left(P_5/P_4\right)^{(0.55)(0.60)} = \left(P_5/P_4\right)^{0.33}$ |
| 2nd rotor | $\left(P_5/P_4\right)^{(0.55)(0.40)} = \left(P_5/P_4\right)^{0.22}$ |

The following equation was derived from the continuity of flow:

$$\dot{m}_g \frac{\sqrt{T_{in}}}{P_{in}} = CA\left(\frac{P_{out}}{P_{in}}\right)^{\frac{1}{k}}\sqrt{1-\left(\frac{P_{out}}{P_{in}}\right)^{\left(\frac{k-1}{k}\right)}}, \qquad [2.7]$$

where k is the ratio of specific heats, C is a proportionality constant, and A is an area. This equation expresses the flow through each of the turbine restrictions. The flow through the first diaphragm is used to determine the mass flow through the turbine with $P_{out}/P_{in}$ being equal to $(P_5/P_4)^{0.27}$.

**2.3.5   Turbine Torque.** The turbine torque equation also had to be determined without the aid of turbine maps. The equation was established from the impulse and momentum equation for the rotor. The Torque on the turbine was expressed as:

$$\text{Torque} = f(\dot{m}_5, V_{gas}, V_{rotor}) \tag{2.8}$$

The velocity of the gas was assumed to be approximately twice the velocity of the rotor. The velocity of the gas was determined at each point through the turbine in order to determine an average velocity. This average velocity and the following equation were then used to calculate gas velocity.

$$V_{gas} = K\sqrt{T_4\left[1-\left(\frac{P_5}{P_4}\right)\right]^{0.073}} \tag{2.9}$$

These assumptions were then combined to determine turbine torque.

**2.3.6   Turbine Temperature Drop.** The equation for the turbine temperature drop was determined in much the same manner as the compressor equation. The coefficient of specific heat, $c_{p_t}$, was considered to be 0.26 at the temperature in the turbine. The equation describing the turbine temperature drop is given by:

$$(T_4 - T_5) = \frac{K\tau_t N}{c_{p_t}\dot{m}_4} \tag{2.10}$$

25

**2.3.7  Torque Balances.** The speed of the engine is determined by integrating the acceleration. The acceleration was established from the relationship:

$$\dot{N} = \frac{(\tau_t - \tau_c)}{KJ} \qquad [2.11]$$

The rotor inertia was estimated from experimental test data.

**2.3.8  Nozzle.** The flow through the exhaust nozzle was determined from the continuity equation for the nonchoked condition similar to equation [2.7]. A limit of 0.534 was placed on the $(P_0/P_7)$ value used in this equation to account for the choked condition.

**2.3.9  Airflow Balances.** In order to relate the airflow through the compressor, turbine and nozzle, airflow balance equations were required. The transient effect was taken into account by considering the compressibility. The general equation used for generating the pressures using compressibility is,

$$\frac{V_c \dot{P}_c}{RT_c} = \dot{m}_{in} - \dot{m}_{out} \qquad [2.12]$$

This equation was used for airflow balances within the combustor and the tailpipe. In addition, the turbine exit pressure $(P_5)$ was determined from the nozzle inlet pressure $(P_7)$. It is assumed that there is a two-percent loss in the tail section. In a similar manner compressor exit pressure $(P_3)$ is determined from turbine inlet pressure $(P_4)$. It is assumed that there is a four- percent loss in the combustor.

All these equations were then used to build a nonlinear engine model within the Simulink environment (Mathworks, Inc., 1994). Figure 2.6 show the block diagram representation of the engine. In this model each of the major components are separately modeled the communication loop and the input and output management blocks handle the passing of variables between the components. This model will be used in subsequent areas for nonlinear evaluation of the work described herein.
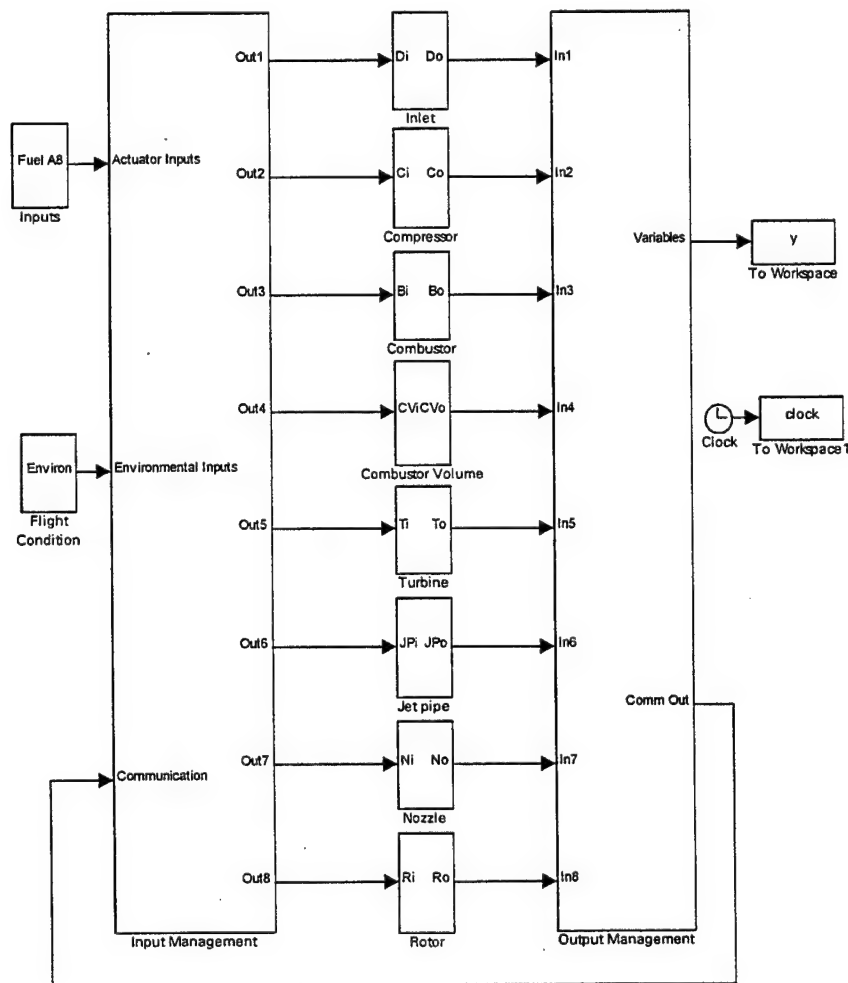


**Figure 2.6 Simulink Block Diagram Representation of the Engine**

## 2.4    Linear Model Generation

In many engineering disciplines a physical process can be modeled by a set of nonlinear differential equations of the form:

$$\dot{x} = f(x,\ u,\ t)$$
$$y = g(x,\ u,\ t)$$

[2.13]

where x is the vector of state variables, y is a vector of outputs, u is a vector of external inputs, t is the scalar time variable and f and g are functionals.  These equations are derived using physical laws and are then nondimensionalized so as to minimize the number of free parameters.

Since many processes are "weakly" nonlinear, they may be approximated by a set of linear, time invariant (LTI) differential equations, called state space models, given by:

$$\dot{x} = A(\alpha)x + B(\alpha)u$$
$$y = C(\alpha)x + D(\alpha)u$$

[2.14]

where $\alpha$ is a vector of parameters corresponding to the point of linearization of [2.13], and A, B, C, D are the system matrices.  In the aircraft engine business it is common practice to use state space models (Geyser, 1979) and (Sellers and Daniele, 1975).  These models are designed for the purpose of designing control laws capable of safe operation of the machinery while achieving high performance levels and near optimal efficiency. When an engineering system can be represented as in equation [2.13] a Taylor series expansion is formed at the parameter point $\alpha$ as follows:

28

$$f(\tilde{x}, \tilde{u}, t) = f(\tilde{x}(\alpha), \tilde{u}(\alpha), t_0) + \left.\frac{\partial f}{\partial \tilde{x}}\right|_{\substack{\tilde{x}=\tilde{x}(\alpha)\\ \tilde{u}=\tilde{u}(\alpha)}} (\tilde{x} - \tilde{x}(\alpha)) + \left.\frac{\partial f}{\partial \tilde{u}}\right|_{\substack{\tilde{x}=\tilde{x}(\alpha)\\ \tilde{u}=\tilde{u}(\alpha)}} (\tilde{u} - \tilde{u}(\alpha)) + \dots \quad [2.15]$$

Since the system is assumed to be in steady state with the state $x = x(\alpha)$ and control input $u = u(\alpha)$, the time derivative of the state vector $\partial x / \partial t$ is zero and thus so is the first term of the expansion. The series is then truncated after the linear terms, resulting in the linear system of equation [2.14]. Here, the change of variable $x = \tilde{x} - \tilde{x}(\alpha)$ and $u = \tilde{u} - \tilde{u}(\alpha)$ is used for ease of notation.

When an engineering system is not represented by a set of nonlinear differential equations, but instead is modeled using detailed computer routines then a perturbation technique must be used in order to extract linear models. A widely accepted technique was developed (Weinberg, 1975) and involves the computation of the system matrices by successively perturbing the model states and measuring the responses within the computer code. In fact such a routine is built into the Matlab/Simulink environment to generate linear models. This functionality of Matlab was used to create the linear models used in this work. The nonlinear model is trimmed at some operating point $\{X_0, U_0\}$, then the linear model will be a perturbation model of the following form:

$$\Delta \dot{X} = A\, \Delta X + B\, \Delta U$$
$$\Delta Y = C\, \Delta X + D\, \Delta U \qquad\qquad [2.16]$$

29

where [A, B, C, D] are the partial derivatives of the state-derivatives with respect to the states and the inputs. It is traditional to drop the $\Delta$ from $\Delta X$, $\Delta Y$ and $\Delta U$ (note that $\Delta U = U - U_0$, etc.) to obtain the linear state-space equations for the system.

In order, to ensure sound numerical properties of the linearized models, the state space system is normalized using a scaling matrix, $R_x$, and a new vector of state variables z as follows:

$$x = R_x z \qquad [2.17]$$

where:

$$R_x = \begin{bmatrix} r_{x1} & 0 & 0 \\ 0 & r_{x2} & 0 \\ 0 & 0 & r_{x3} \end{bmatrix}, \qquad [2.18]$$

and $r_{x1}$, $r_{x2}$ and $r_{x3}$ are the $\Delta$'s (perturbation amounts) used to create the linear models. In addition, the inputs and outputs are scaled by their perturbation values, defining a normalized input vector $\tilde{u}$ and normalized output vector $\tilde{y}$ defined by:

$$u = R_u \tilde{u}, \; y = R_y \tilde{y} \qquad [2.19]$$

where the diagonal matrices $R_u$ and $R_y$ contain the perturbation amounts ($\Delta$'s) for the inputs and outputs, respectively. The state space model then becomes:

30

$$\dot{z} = R_x^{-1}A(\alpha)R_x z + R_x^{-1}B(\alpha)R_u \tilde{u}$$
$$\tilde{y} = R_y^{-1}C(\alpha)R_x z + R_y^{-1}D(\alpha)R_u \tilde{u}$$

[2.20]

or simply:

$$\dot{z} = \widetilde{A}(\alpha)z + \widetilde{B}(\alpha)\tilde{u}$$
$$\tilde{y} = \widetilde{C}(\alpha)z + \widetilde{D}(\alpha)\tilde{u}$$

[2.21]

## 2.5    Plant Set Selection

Plant sets considered for robust controller design must satisfy two conditions. First, they must satisfy the path connectedness condition. Secondly, the plant outputs must exhibit a reasonable level of response from their associated inputs over the entire region being considered. Evaluation of the sea level linear models from idle to 100% corrected speed the thrust response to fuel flow input was reasonable. However, the stall margin output to nozzle area input had two distinct response characteristics, this was attributed to the choking and unchoking of the nozzle. This means that the region for control design had to be further reduced to include only the choked nozzle region of operation, which is approximately 50% power to 100% power level.

31

# CHAPTER 3

## TRACKING FILTER DESIGN

Model-based control will not work well without first having a good model of the plant (Adibhatla and Gastineau, 1994). Any model, from linear models to piecewise linear models to nonlinear models, can be used. However, to obtain the accuracy and flexibility required for direct thrust control, it is useful to model the physics of the engine accurately. Also, any model that is created will represent some nominal plant. Therefore, a means of changing the model so that it will account for plant deterioration with age, manufacturing differences, and modeling errors is required. This is the function of the tracking filter. Simply put, a tracking filter adjusts the model's inputs and parameters to make it match the actual plant outputs. The tracking filter works by forcing a selected set of model outputs to match the corresponding plant sensors by adding biases to selected model inputs or parameters. The design is further complicated when the plant can be operated or used in more than one way. The desire is that one tracking filter can be found that allows the plant to operate in these different modes. Unfortunately to date such a filter has not yet been found.

## 3.1    Tracking Filter Selection

The tracking filter is the heart of any model-based control system. How well it is designed will have a direct impact on how well the entire control system will perform. At this point it is important for the engineer to decide how the model-based control features will be used. Since every variable is now available because of the model it appears that the design is simplified. This is not the case, due to the fact that the number of

32

parameters that may be updated or adjusted is much larger than the number of values that can be sensed from the actual plant. The fact is that the difficulty of designing a control has been shifted from the closed loop controller design to the design of the tracking filter.

The structure of the tracking filter and model loop is nonlinear. This presents some special design problems due to the fact that there is a lack of design techniques. In order to solve this problem as suggested by Misawa and Hedrick (1989), the nonlinear model will be linearized then the design will proceed to find an acceptable design for every plant in the plant set. This suggests that the use of loop shaping techniques would be appropriate for the tracking filter design.

If the designer wants to use a state-variable method and if modeling errors exist, or the plant is subjected to unknown disturbances, the plant and model states will be different in the steady state. As a result, the corresponding outputs will deviate from each other. In this circumstance the Proportional-plus-Integral (PI) Observer is considered to be appropriate. Before studying PI observers, first consider the development of a nonlinear observer in an attempt to identify the key issues that must be addressed in any design. Essentially, it is postulated that a dynamic system exists that has the property that the state of the observer converges to the state of the observed process. The desire is to find this dynamical system.

## 3.2    Nonlinear Observers

An observer for a plant, consisting of a dynamic system

$$\dot{x} = f(x, u) \qquad\qquad [3.1]$$

with observations given by

$$y = g(x, u) \qquad\qquad [3.2]$$

is another dynamic system, the state of which is denoted by $\hat{x}$, excited by the output y of

the plant, having the property that the error

$$e = x - \hat{x} \qquad\qquad [3.3]$$

converges to zero in the steady state.

One way of obtaining an observer is to imitate the procedure used in a linear

system, namely to construct a model of the original system given in equation [3.1] and

force it with the residual:

$$r = y - \hat{y} = y - g(\hat{x}, u) \qquad\qquad [3.4]$$

The equation for a proportional observer becomes

$$\dot{\hat{x}} = f(\hat{x}, u) + k(y - g(\hat{x}, u)) \qquad\qquad [3.5]$$

where k( ) is a suitably chosen nonlinear function. The differential equation for the error

$\dot{e}$ can be used to study its behavior. This equation is given by

$$
\begin{aligned}
\dot{e} &= \dot{x} - \dot{\hat{x}} \\
&= f(x, u) - f(\hat{x}, u) - k\big(g(x, u) - g(\hat{x}, u)\big) \qquad\qquad [3.6] \\
&= f(x, u) - f(x - e, u) + k\big(g(x - e, u) - g(x, u)\big)
\end{aligned}
$$

Suppose that by the proper choice of k( ) the error equation [3.6] can be made asymptotically stable, so that an equilibrium state is reached for

$$\dot{e} = 0$$

Then in equilibrium, equation [3.6] becomes

$$0 = f(x,u) - f(x-e,u) + k\big(g(x-e,u) - g(x,u)\big) \qquad [3.7]$$

Since the right-hand side of equation [3.7] becomes zero when e = 0, independent of x and u, it is apparent that e = 0 is an equilibrium state of equation [3.6]. This implies that if the k( ) can be chosen to achieve asymptotic stability, the estimation error e converges to zero.

It is very important to notice that the right-hand side of equation [3.7] becomes zero independent of x and u only when the nonlinear functions $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ used in the observer are exactly the same as in the equations [3.1] and [3.2] that define the plant dynamics and observations, respectively. Any discrepancy between the corresponding functions will generally prevent the right-hand side of equation [3.7] from vanishing and hence will lead to a steady-state estimation error. Since the mathematical model of the physical process is always an approximation, in practice the steady-state estimation error will generally not go to zero. But, by careful modeling, the discrepancies can usually be minimized between the f and g functions of the true plant and the model used in the observer. This can keep the steady-state estimation error acceptably small.

35

For the same reason, that the model of the plant and the observation that is used in the observer must be accurate, it is important that the control signal u that goes into the plant is the very same control used in the observer. If the control to the plant is subject to saturation, for example, then the nonlinear function that models the saturation must be included in the observer. Failure to observe this precaution can cause difficulties. This problem is overcome by using actuator feedback values as the inputs into the model, thereby, accounting for any nonlinear behavior out of the actuators.

## 3.3    Proportional Plus Integral Observer

Here we study the proportional-plus-integral (PI) observer that was first introduced by Wojciechowski (1978) for single-input, single-output systems and further developed for multivariable systems by Kaczorek (1979) and Shafai (1985). Finally, Beale and Shafai (1989) present an investigation of the robustness properties of a PI observer used in a feedback compensation configuration. The PI observer offers the designer additional degrees of freedom to actually get the desired response.

Consider a linear time-invariant system described by

$$
\begin{aligned}
\dot{x} &= Ax(t) + Bu(t) \\
y &= Cx(t)
\end{aligned}
\qquad [3.8]
$$

where the state x, the input u and the output y are n-, m- and p-dimensional vectors, respectively. Without loss of generality assume that

36

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1p} \\ \vdots & \ddots & \vdots \\ A_{p1} & \cdots & A_{pp} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 \\ \vdots \\ B_p \end{bmatrix}$$

$$C = \begin{bmatrix} C_1 & \cdots & C_p \end{bmatrix}$$

[3.9]

where

$$A_{ii} = \begin{bmatrix} 0 & 0 & \cdots & 0 & x \\ 1 & 0 & \cdots & 0 & x \\ 0 & 1 & \cdots & 0 & x \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & x \end{bmatrix}, \quad A_{ij} = \begin{bmatrix} 0 & \begin{matrix} x \\ x \\ x \\ \vdots \\ x \end{matrix} \end{bmatrix}$$

[3.10a]

$$C_i = \begin{bmatrix} 0 & \begin{matrix} 0 \\ 0 \\ \vdots \\ 1 \\ x \\ \vdots \\ x \end{matrix} \end{bmatrix} \leftarrow \text{row } i \qquad i,j = 1,\ldots,p$$

[3.10b]

and B has no special form. Note that the non-zero entries are denoted by x and $A_{ii}$, $A_{ij}$, $C_i$, are $v_i \times v_i$, $v_i \times v_j$, $p \times v_i$ submatrices with $v_i$ being the observability indices of the system.

It is known that any observable linear time-invariant system of the form given in equation [3.8] that is not in the observable canonical form can be transformed by an equivalence relation to the structure given by equation [3.9], (Kailath, 1980, pp. 424-437). The methodology for finding this equivalence relation is given in Appendix A.

37

Now, consider a linear time-invariant system

$$\dot{\hat{x}}(t) = (A - GC)\hat{x}(t) + Gy(t) + Bu(t) + Kw(t)$$
$$\dot{w}(t) = y(t) - C\hat{x}(t)$$
[3.11]

where $\hat{x}(t)$ and $w(t)$ are n- and p- dimensional vectors, respectively. G and K are matrices of the proportional-plus-integral observer that must be determined.

Definition 3.1   The system described by equation [3.11] is said to be a full-order PI observer for the system given in equation [3.8] if and only if:

$$\lim_{t \to \infty} e(t) = 0$$
$$\lim_{t \to \infty} w(t) = 0$$
[3.12]

where $e(t) = \hat{x}(t) - x(t)$ represents the observer error.

Theorem 3.1   (Beale and Shafai, 1989)  The system given in equation [3.11] is a full-order PI observer for the system given in equation [3.8] if and only if all the eigenvalues of the matrix:

$$R = \begin{bmatrix} A - GC & K \\ -C & 0 \end{bmatrix}$$
[3.13]

have negative real parts.

*Proof:* From equations [3.8] and [3.11] the dynamics of the observer error becomes:

$$\dot{e}(t) = (A - GC)e(t) + Kw(t)$$
[3.14]

38

which yields

$$\begin{bmatrix} \dot{e}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} A - GC & K \\ -C & 0 \end{bmatrix} \begin{bmatrix} e(t) \\ w(t) \end{bmatrix},$$  [3.15]

and it follows immediately that $\hat{x}(t)$ is an estimate of $x(t)$ if and only if Re $\lambda_q[R] < 0$

for $q = 1, \ldots, (n + p)$.

Thus the problem reduces to one of choosing the matrices G and K such that the system given in equation [3.11] will be a PI observer for the system given in equation [3.8]. A methodology for selecting the matrices G and K for both single-output and multiple-output systems is known. These methodologies were presented by Beale and Shafai (1989).

## 3.4    Proportional-Plus-Integral Observer Design for Single-Output Systems

In this case $p = 1$ and the system matrices have the form

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}, \quad B = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

[3.16]

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Let the unknown PI observer matrices G and K be written as

$$\begin{bmatrix} G \\ \hline 1 \end{bmatrix} \begin{pmatrix} n \\ \end{pmatrix}$$  [3.17]

Then a simple algebraic manipulation shows that

$$\det\left(\lambda I_{(n+1)} - R\right) = \det\begin{bmatrix} \lambda I_n - (A - GC) & -K \\ C & \lambda \end{bmatrix}$$
$$= \lambda^{(n+1)} + \sum_{i=1}^{n}\left(g_{(i-1)} + a_{(i-1)}\right)\lambda^i + \sum_{i=0}^{(n-1)}k_i\lambda^i$$  [3.18]

and by comparison of its coefficients with the ones specified by the desired characteristic equation, i.e.

$$\Delta_d(\lambda) = \prod_{i=1}^{(n+1)}(\lambda - \lambda_i)$$  [3.19]

this results in a system of (n + 1) equations, which have to be satisfied by 2n elements of G and K. Therefore, (n - 1) elements of G and K can be chosen arbitrarily (except $g_{(n-1)}$ and $k_0$) and the remaining elements can be evaluated from the system of (n + 1) equations.

40

## 3.5    Proportional-Plus-Integral Observer Design for Multiple-Output Systems

In the general case of p outputs, we decompose the system given in equation [3.8]

into p subsystems each having dimension $v_i$ as specified by equations [3.9] and [3.10].

Let the state vector of each subsystem be represented by

$$x_i = \begin{bmatrix} x_{\sigma_{(i-1)}+1} \\ x_{\sigma_{(i-2)}+1} \\ \vdots \\ x_{\sigma_i} \end{bmatrix}, \quad i = 1, \ldots, p \qquad [3.20]$$

where

$$\sigma_i = \sum_{j=1}^{i} v_j \qquad [3.21]$$

Then the ith subsystem can be written as

$$\dot{x}_i = A_{ii}x_i + \sum_{\substack{j=1 \\ j \neq i}}^{p} A_{ij}x_j + B_i u \qquad [3.22]$$

or equivalently as

$$\dot{x}_i = A_{ii}x_i + \sum_{\substack{j=1 \\ j \neq i}}^{p} a_{i\sigma_j}x_{\sigma_j} + B_i u \qquad [3.23]$$

41

where $A_{ii}$, $A_{ij}$, and $B_i$ are the submatrices defined by equations [3.9] and [3.10], and $a_{i\sigma j}$ is

the $\sigma_j$th column of the matrix $A_{ij}$. Also let $C_\sigma$ be the pxp lower triangular matrix formed

from C by eliminating all by the p non-trivial columns of C so that we have

$$C_\sigma = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ x & 1 & \cdots & 0 \\ x & x & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ x & x & \cdots & 1 \end{bmatrix} \qquad [3.24]$$

Using [3.8] we generate a new output equation as

$$\tilde{y} = C_\sigma^{-1} y = \begin{bmatrix} x_{\sigma_1} \\ x_{\sigma_2} \\ \vdots \\ x_{\sigma_p} \end{bmatrix} \qquad [3.25]$$

so that the $X_{\sigma i}$, $i = 1, ..., p$, are available. Thus, it is apparent that the ith subsystem

described by equation [3.23] is a $v_i$-dimensional single-output system in observable

companion form driven by the directly measurable system signals u(t) and $x_{\sigma_j}$ (j $\neq$ i).

Therefore, the procedure, which was given for single-output systems, can now be

employed for each of the p subsystems.

## 3.6 Linear Versus Nonlinear Tracking Filter

The design of a nonlinear tracking filter is possible. It would most likely be based

upon neural networks or fuzzy logic techniques. These techniques have recently gained

favor within the control community. But, the amount of data necessary to design a neural network was considered prohibitively large for the task that needed to be accomplished. For this work a linear tracking filter was deemed sufficient for the following reasons. First, the dynamics of the engine change slowly as it is flown throughout the flight envelope. Secondly, the error between the actual plant and the model will in general be small. This is due to the amount of time developing the model during the engine development process. This small error means that any correction to the model will adjust the model outputs in a linear fashion. Finally, linear observer techniques are better understood and offer a satisfactory solution for this problem.

If the plant had been highly nonlinear or the dynamic behavior had changed rapidly then the only choice may have been to design a nonlinear tracking filter. Also, if the model contains first principle physics of the plant and is not necessarily corrected or adjusted based upon development testing results then nonlinear techniques may eliminate or reduce the development time required for the model.

# CHAPTER 4

## QUANTITATIVE NYQUIST ARRAY USING
## FORWARD PATH DECOUPLING

The idea of interaction between control loops is of paramount concern when developing control systems for multiple-input, multiple-output systems. For any practical design to be decoupled or to have very little interaction between loops is highly desirable (Suh, 1990). In this chapter, a methodology for reducing loop interaction is presented and then tied to a quantitative design for a diagonal feedback controller.

After the tracking filter has been successfully designed and tested to meet specifications one can now consider the n-input, n-output feedback control system shown in Figure 4.1. In a model-based control structure, the dynamics of the tracking filter and model are now included in the definition of the plant P. The feedback values are now model computed values.
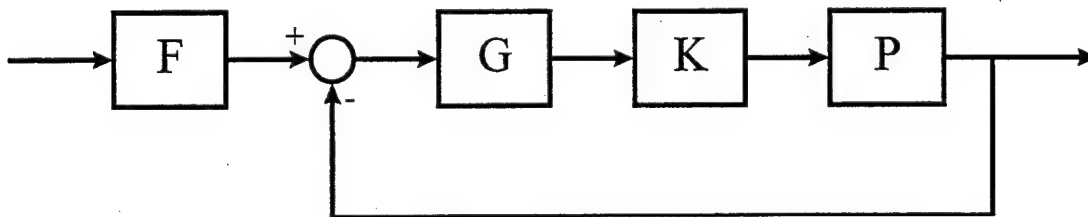


**Figure 4.1 Feedback Structure**

The transfer function for the system shown above is given by:

$$T_{Y/R} = (I + PKG)^{-1} PKGF$$
$$= (\hat{P} + KG)^{-1} KGF$$

[4.1]

where $\hat{P} \equiv P^{-1}$ and has the elements $\hat{p}_{ij} = \dfrac{1}{q_{ij}}$.

The problem that is being considered here is the quantitative feedback design problem specified by Horowitz (1979) of designing a decoupling, precompensating matrix, K, with diagonal elements equal to unity ($k_{ii} = 1$), a diagonal feedback controller, G, and a prefilter, F, to achieve certain tracking specifications given by:

$$A_{ij}(\omega) \leq T_{Y/R}(j\omega) \leq B_{ij}(\omega) \qquad i,j = 1, 2, \dots n$$

[4.2]

$A_{ij}(\omega)$ and $B_{ij}(\omega)$ are the client's desired response characteristics, for all plants, $P \in P$, where due to uncertainty, $P$ is the set of all possible plants. All matrices are nxn.

The structure of Figure 4.1 was selected over the structure used by Nordgren, et.al. (1994) because of the apparent ease of implementation. The approach used by Nordgren emphasized the ease of design over implementation. However, it is important to note that the level of performance achievable by the two configurations is identical.

45

## 4.1    Preliminary Mathematics

The following is a collection of definitions and elementary results that will be used for the work contained herein.  It is based on the theory of non-negative matrices (Berman and Plemmons, 1979).

*Definition 4.1*:  A matrix, $M \in R^{n \times n}$ is an M-matrix if the diagonal elements of M are

positive, the off diagonal elements of M are non-positive, and the

principal minors of M are non-negative.

*Definition 4.2*:  Given a matrix $Z \in C^{n \times n}$ the comparison matrix of Z, M(Z), has elements,

$$M(Z)_{ij} = \begin{cases} |z_{ii}| & \text{(diagonal elements)} \\ -|z_{ij}| & i \neq j \end{cases}$$

*Definition 4.3*:  A matrix, $Z \in C^{n \times n}$ is irreducible if there does not exist a permutation,

$P \in C^{n \times n}$ such that

$$PZP^{-1} = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}, \text{ with square submatrices, } Z_{11} \text{ and } Z_{22}.$$

*Definition 4.4*:  An irreducible matrix, $Z \in C^{n \times n}$ is an H-matrix (Hadamard matrix) if

$M(Z)$ is an M-matrix.

*Definition 4.5*:  An H-matrix, $Z \in C^{n \times n}$ is a BH-matrix if $Z^{-1}$ is also an H-matrix.

*Definition 4.6*:  A matrix, Z admits a diagonal regular splitting if it can be written as

$Z = D + C$, with $D = \text{diag}\{Z\}$ non-singular.

*Definition 4.7*: The Perron root of a matrix M, is the largest eigenvalue of the matrix

formed from the absolute values of the elements of M.

$$\lambda_p(M) = \max\{\lambda(M_+)\} = \rho(M_+)$$

$\rho(M_+)$ is the spectral radius of $M_+$. The absolute values are taken

element-wise, $(M_+)_{ij} = |M_{ij}|$.

The Perron root is analytic and monotonic with respect to the elements

of $M_+$. The Perron root is an upper bound on the eigenvalues, $\lambda(M)$, and

therefore the spectral radius, $\rho(M)$, of a matrix:

$$|\lambda(M)| \leq \rho(M) \leq \lambda_p(M)$$

The spectral radius is a lower bound on all induced norms.

*Definition 4.8*: Given that $Z \in C^{n \times n}$ admits the diagonal splitting,

$$Z = D + C = (I + CD^{-1})D = (I + M)D$$

M is the interaction matrix and the interaction index of Z is the Perron

root of the interaction matrix,

$$\gamma(Z) = \lambda_p(M)$$

47

The utility of the interaction index for feedback design is seen by considering the inverse

of Z, $\left(Z^{-1} = \hat{Z}\right)$. If Z is an H-matrix, M has spectral radius less than unity

$\left(\rho(M) \le \lambda_p(M) < 1, \text{ with } \gamma(Z) \equiv \lambda_p(M)\right)$ and $(I+M)^{-1} = \sum_{k=0}^{\infty}(-M)^k$ converges.

Also,

$$\lambda_p(\hat{Z}) = \lambda_p\left(D^{-1}(I+M)^{-1}\right) = \lambda_p\left(\sum_{k=0}^{\infty}(-M)^k\right) \le \sum_{k=0}^{\infty}\left(\lambda_p(M)\right)^k = \frac{\lambda_p(Z)}{1-\lambda_p(Z)} \qquad [4.3]$$

*Definition 4.9*: An irreducible matrix $Z \in C^{n \times n}$ is almost decoupled if the

$\max\left\{\lambda_p(Z), \lambda_p(\hat{Z})\right\} \le 0.5$. (Z almost decoupled $\Rightarrow$ Z is a BH matrix -

see below.)

*Theorem 4.1*: An irreducible matrix, $Z \in C^{n \times n}$ is a BH-matrix if

$\min\left\{\lambda_p(Z), \lambda_p(\hat{Z})\right\} \le 0.5$. Notice that the concept of almost decoupling

is stronger than the requirements for Z to be a BH-matrix.

*Proof*: See (Nwokah, et al, 1995) or appendix B.

*Theorem 4.2*: An H-matrix $Z \in C^{n \times n}$ is almost decoupled, if $\lambda_p(\hat{Z}) \le \dfrac{\lambda_p(Z)}{1-\lambda_p(Z)} \le 0.5$,

requiring $\lambda_p(Z) \le 1/3$.

*Proof*: This proof follows from equation [4.3] and definition 4.9.

48

*Theorem 4.3*: Given $\hat{P} \in \hat{\mathcal{P}}$, square and invertible with no hidden unstable modes, the

system,

$$
\begin{aligned}
T_{Y/R} &= (I + PKG)^{-1}PKGF \\
&= (\hat{P}_D + G)^{-1}\left(I + (\hat{P}_O + (K-I)G)(\hat{P}_D + G)^{-1}\right)^{-1}KGF \\
&= (\hat{P}_D + G)^{-1}(I + M)^{-1}KGF
\end{aligned}
\qquad [4.4]
$$

is internally stable if:

i)      G, F, K are designed to be stable.

ii)     $\lambda_p(M) < 1.$

iii)    Each $(1 + g_I q_{ii})$ is designed to have no zeros in the right hand plane

For proof see appendix B.

## 4.2    Insights of Interaction in Multiple-Input, Multiple-Output Systems

The idea of being decoupled or having very little interaction between loops is highly desirable in any practical design (Suh, 1990). The goals of decoupling can be illustrated as follows. Consider the return difference matrix and its regular splitting into a diagonal matrix D and an off-diagonal matrix C:

$$
(I + PKG) = D + C = (I + CD^{-1})D
\qquad [4.5]
$$

49

The goal of decoupling is to force the spectral radius ($\rho$) of the modulus of the product $CD^{-1}$ to be less than one. In this way, the diagonal elements of the return difference matrix solely determine the closed loop stability of the system.

Methods such as the Relative Gain Array (RGA) are used to measure the interaction between inputs and outputs. It is shown in Section 4.1 that the Perron root can be a useful measure of the degree of interaction in a feedback design. It is important to point out that the interaction index is a measure of generalized diagonal dominance rather than strict diagonal dominance of a matrix.

## 4.3    Precompensation Matrix Design

Consider equation [4.1] and separate it so that the diagonal and off-diagonal components are identified.

$$\begin{aligned}
T_{Y/R} &= \left(\hat{P} + KG\right)^{-1} KGF \\
&= \left(\hat{P}_D + \hat{P}_O + (K-I)G + G\right)^{-1} KGF \\
&= \left(\hat{P}_D + G + \hat{P}_O + (K-I)G\right)^{-1} KGF \\
&= \left(\hat{P}_D + G\right)^{-1}\left(I + \left(\hat{P}_O + (K-I)G\right)\left(\hat{P}_D + G\right)^{-1}\right)^{-1} KGF \qquad [4.6]
\end{aligned}$$

Notice that equation [4.6] is in the form of equation [4.4] where:

$$C = \hat{P}_O + (K-I)G$$
$$D = \left(\hat{P}_D + G\right)$$

50

therefore;

$$M = CD^{-1} = \left(\hat{P}_O + (K-I)G\right)\left(\hat{P}_D + G\right)^{-1}.$$ [4.7]

Now for a given plant set it is necessary to determine how to design K so that the interaction index, $\gamma = \lambda_p(M)$, is less than unity. First, define the diagonal sensitivity (S) and diagonal complementary sensitivity (T) as:

$$S = \left(I + G\hat{P}_D^{-1}\right)^{-1}, \; s_{ii} = \frac{1}{1 + g_i q_{ii}} = \frac{1}{1 + \ell_i}$$ [4.8]

$$T = G\left(\hat{P}_D + G\right)^{-1}, \; t_{ii} = \frac{g_i q_{ii}}{1 + g_i q_{ii}} = \frac{\ell_i}{1 + \ell_i}$$ [4.9]

The perron root is found by solving the maximum eigenvalue-eigenvector problem of equation [4.7] which is as follows.

$$\left|\left(\hat{P}_O + (K-I)G\right)\left(\hat{P}_D + G\right)^{-1}\right| x = \lambda_p x$$

$$\left|\hat{P}_O\left(\hat{P}_D + G\right)^{-1} + (K-I)G\left(\hat{P}_D + G\right)^{-1}\right| x = \lambda_p x$$

$$\left|\hat{P}_O\left(\left(I + G\hat{P}_D^{-1}\right)\hat{P}_D\right)^{-1} + (K-I)G\left(\hat{P}_D + G\right)^{-1}\right| x = \lambda_p x$$

$$\left|\hat{P}_O\hat{P}_D^{-1}\left(I + G\hat{P}_D^{-1}\right)^{-1} + (K-I)G\left(\hat{P}_D + G\right)^{-1}\right| x = \lambda_p x$$ [4.10]

Substituting equations [4.8] and [4.9] into equation [4.10] the following result is obtained.

51

$$\left| \hat{P}_O \hat{P}_D^{-1} S + (K - I) T \right| x = |N| x = \lambda_p x \qquad [4.11]$$

with the elements of N given by:

$$|n_{ij}| = \left| \frac{q_{jj}}{q_{ij}} \frac{1}{1 + \ell_j} + k_{ij} \frac{\ell_j}{1 + \ell_j} \right|$$

$$= \left| \left( \frac{q_{jj}}{q_{ij}} - k_{ij} \right) \frac{1}{1 + \ell_j} + k_{ij} \right| \qquad [4.12]$$

If only the specification for the magnitude of the sensitivity of individual loops is known, i.e. $\left| \dfrac{1}{1 + \ell_j} \right| \leq s_j(\varpi)$, we may apply the Schwartz inequality to obtain the following.

$$|n_{ij}| \leq \left| \left( \frac{q_{jj}}{q_{ij}} - k_{ij} \right) s_j + k_{ij} \right| \leq \left| \left( \frac{q_{jj}}{q_{ij}} - k_{ij} \right) s_j \right| + |k_{ij}| \qquad [4.13]$$

To guarantee reduction of the interaction index using the precompensating matrix, K, which has only off-diagonal elements for design, $|n_{ij}|$ may be reduced by designing

$$\left| \frac{q_{jj}}{q_{ij}} - k_{ij} \right| \leq \left| \frac{q_{jj}}{q_{ij}} \right| \text{ or }$$

$$\frac{\max}{P \in P} \left| 1 - k_{ij} \frac{q_{ij}}{q_{jj}} \right| \leq 1. \qquad [4.14]$$

52

This can be achieved over frequency using a QFT type design. However, from equation [4.12] the following ideas need to be considered when designing the matrix K. First, when $|\ell_j| << 1$ we find that $|n_{ij}| \approx |q_{jj}/q_{ij}|$ so that the interaction index is the same as that of the open loop plant. Secondly, when $|\ell_j| >> 1$ then we find that $|n_{ij}| \approx |k_{ij}|$. This motivates choosing the off-diagonal elements of K as small as possible. Finally, the third consideration is that when $|\ell_j| \approx 1$, especially around the gain and phase cross over frequencies of $\ell_j$ where typically a margin such as $\left| \dfrac{1}{1+\ell_j} \right| \leq \beta$, $\beta > 0$ dB is specified, then selecting $k_{ij} \approx \dfrac{q_{jj}}{q_{ij}}$ will keep individual elements of $|N|$ small.

## 4.4    Diagonal Loop Controller Design

There exist several methodologies for a multivariable QFT design. The technique used here makes use of the plant inverse. Now consider equation [4.1], after the precompensating matrix, K, has been designed it is easier to rework the equation into the following form.

$$
\begin{aligned}
T_{Y/R} &= \left(\hat{P} + KG\right)^{-1} KGF \\
&= \left(\hat{K}\hat{P} + G\right)^{-1} GF
\end{aligned}
\qquad [4.15]
$$

Then defining $\left(\hat{K}\hat{P}\right)_{ij} \equiv \left(\hat{P}^*\right)_{ij} = \dfrac{1}{q_{ij}^*}$, one has the standard QFT design problem given by:

53

$$T_{Y/R} = \left(\hat{P}^* + G\right)^{-1} GF \qquad [4.16]$$

Using simple algebraic manipulations equation [4.16] is rearranged into the following form, then specific equations are derived for each $t_{ij}$ in the matrix of transfer functions.

$$\left(\hat{P}^* + G\right)T_{Y/R} = GF \qquad [4.17]$$

In order to illustrate, consider the 2x2 system with G diagonal and F fully populated. Expanding equation [4.17] in detail one gets:

$$\left(\begin{bmatrix} \dfrac{1}{q_{11}^*} & \dfrac{1}{q_{12}^*} \\ \dfrac{1}{q_{21}^*} & \dfrac{1}{q_{22}^*} \end{bmatrix} + \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}\right)\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} = \begin{bmatrix} g_1 f_{11} & g_1 f_{12} \\ g_2 f_{21} & g_2 f_{22} \end{bmatrix} \qquad [4.18]$$

Equation [4.18] contains four multiple-input, single-output equations given by:

$$\left(\frac{1}{q_{11}^*} + g_1\right)t_{11} + \frac{1}{q_{12}^*}t_{21} = g_1 f_{11}$$

$$\left(\frac{1}{q_{11}^*} + g_1\right)t_{12} + \frac{1}{q_{12}^*}t_{22} = g_1 f_{12}$$

$$\left(\frac{1}{q_{22}^*} + g_2\right)t_{21} + \frac{1}{q_{21}^*}t_{11} = g_2 f_{21} \qquad [4.19]$$

$$\left(\frac{1}{q_{22}^*} + g_2\right)t_{22} + \frac{1}{q_{21}^*}t_{12} = g_2 f_{22}$$

All four equations can be reduced to the following

54

$$t_{11} = \frac{\ell_1 f_{11}}{1+\ell_1} - \frac{1}{1+\ell_1} \frac{q_{11}^*}{q_{12}^*} t_{21} \qquad t_{12} = \frac{\ell_1 f_{12}}{1+\ell_1} - \frac{1}{1+\ell_1} \frac{q_{11}^*}{q_{12}^*} t_{22}$$

$$t_{21} = \frac{\ell_2 f_{21}}{1+\ell_2} - \frac{1}{1+\ell_2} \frac{q_{22}^*}{q_{21}^*} t_{11} \qquad t_{22} = \frac{\ell_2 f_{11}}{1+\ell_2} - \frac{1}{1+\ell_2} \frac{q_{22}^*}{q_{21}^*} t_{12}$$

[4.20]

where $\ell_i = g_i q_{ii}^*$.

Or generically this can be written as:

$$t_{ij} = \frac{f_{ij}\ell_i}{1+\ell_i} - \frac{1}{1+\ell_i}\left( \sum_{k=1,k\neq i}^{m} \frac{q_{ii}^*}{q_{ik}^*} t_{kj} \right) \quad \text{for } j = 1...m.$$

[4.21]

Now specifications on the individual loop sensitivities can be established to reduce the interaction index to required levels. These specifications would be in addition to the uncertainty reduction and disturbance rejection specifications that are used in standard QFT designs.

## 4.5    Prefilter Design

In order to achieve enhanced or faster response, a prefilter is placed at the input to each channel. The specifications for most QFT designs are generally non-interacting. The feedback controller is designed to achieve specified levels of uncertainty reduction for the diagonal closed loop elements. Then the prefilter would only have diagonal elements designed such that the closed loop response would lie within the client's specifications. If the closed loop system is lower triangular or almost lower triangular then we may be able to further reduce the open loop interaction index by designing off

55

diagonal elements in the prefilter. Consider the following example to illustrate the methodology for designing the prefilter.

Suppose $T = (I + L)^{-1} L = \begin{bmatrix} t_{11} & \varepsilon_{12} \\ t_{21} & t_{22} \end{bmatrix}$, with $|\varepsilon_{ij}| \ll 1$. Choose $F = \begin{bmatrix} f_{11} & 0 \\ f_{21} & f_{22} \end{bmatrix}$ and design

$T_{Y/R} = TF$ in the following order.

1) First design the diagonal elements $f_{ii}$ to achieve the client's tracking specifications on

   $(T_{Y/R})_{ii}$, $i = 1, 2, 3$.

2) For a reduction in interaction,

$$\left| (T_{Y/R})_{21} \right| = \left| t_{21} f_{11} + t_{22} f_{21} \right| \le \left| t_{21} \right|, \text{ or } \left| 1 + \frac{t_{22}}{t_{21} f_{11}} f_{21} \right| \le \left| \frac{1}{f_{11}} \right| \qquad [4.22]$$

Note, in the worst case, the same interaction index can be obtained by choosing $f_{ij} = 0$.

This design can be accomplished using techniques from Quantitative Feedback Theory.

# CHAPTER 5

## DESIGN EXAMPLE

### 5.1    Engine Model

The equations used in Chapter 2 were used to develop a nonlinear turbojet engine simulation model for the purpose of designing a single multivariable controller for an engine at sea level static conditions. This model was then linearized, resulting in thirty-one plant models or sets of state space matrices. Refer to appendix D for this set of normalized plant models. Each of these linearized models was extracted from the nonlinear model using a perturbation technique. Each model consists of two inputs, three states, and either two outputs if designing the controller and precompensating matrix or three outputs if designing the tracking filter. This is summarized in Table 5.1. Also included to improve the realism of the problem were models of the transport, combustion, and sensor time delays, as well as actuator models for nozzle exit area control and fuel flow rates. A schematic diagram of these details is shown in Figure 5.1.

In order to obtain a successful robust controller design, the path connectedness condition must be satisfied, and all the outputs must exhibit a reasonable amount of response due to their associated inputs (Nordgren, 1994). The linear models used for this work were trimmed about values of percent corrected speed that ranged from 70% (idle speed) to 100% (maximum rotation speed). A study of path connectedness was

57

**Table 5.1 Engine Model Variables**

| Input | States | Outputs | |
| --- | --- | --- | --- |
| | | Tracking Filter | Control |
| Fuel Flow | Speed | Speed | Thrust |
| Nozzle Area | Combustor Pressure | Compressor Exit Pressure | Stall Margin |
| | Tailpipe Pressure | Turbine Exit Pressure | |
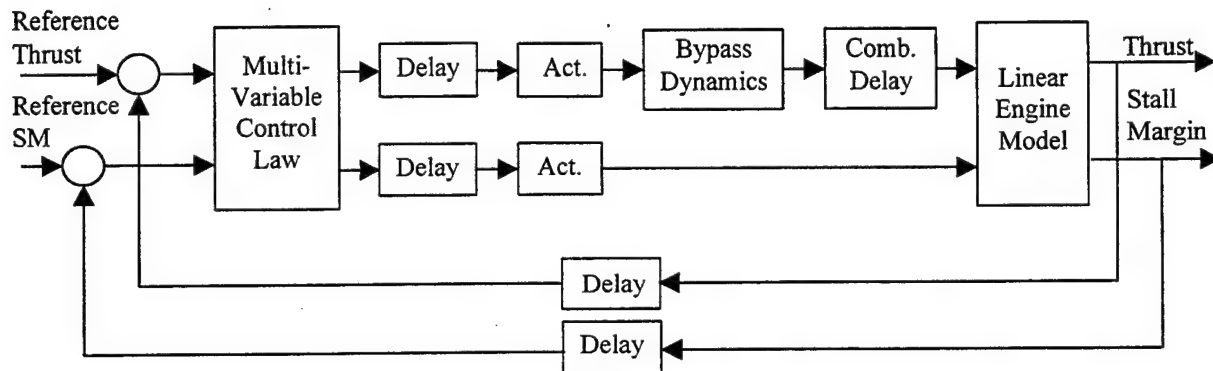


**Figure 5.1 Feedback Configuration Showing Complete Modeling Details After the Tracking Filter Design is Complete.**

performed and plant number one through number twenty-nine were found to be path connected. Closer evaluation of plants thirty and thirty-one revealed that the perturbation levels used to generate the linear models actually resulted in the model interpolating a value for the compressor that was off the map. Therefore, there is not a necessarily smooth connected path set when we move off the maps, this resulted in the elimination of these plants from the acceptable set. This path connectedness study made use of the ratio of the determinants of the plant nominal model and the specific plant of interest. If this

ratio crosses the imaginary axis as a sweep of frequencies is done then the selected plant is not path connected to the nominal plant (Nwokah and Thompson, 1989). In this case the nominal plant is plant number sixteen. Figures 5.2 and 5.3 show plots for a path connected plant and nonpath connected plant respectively.

plant 20
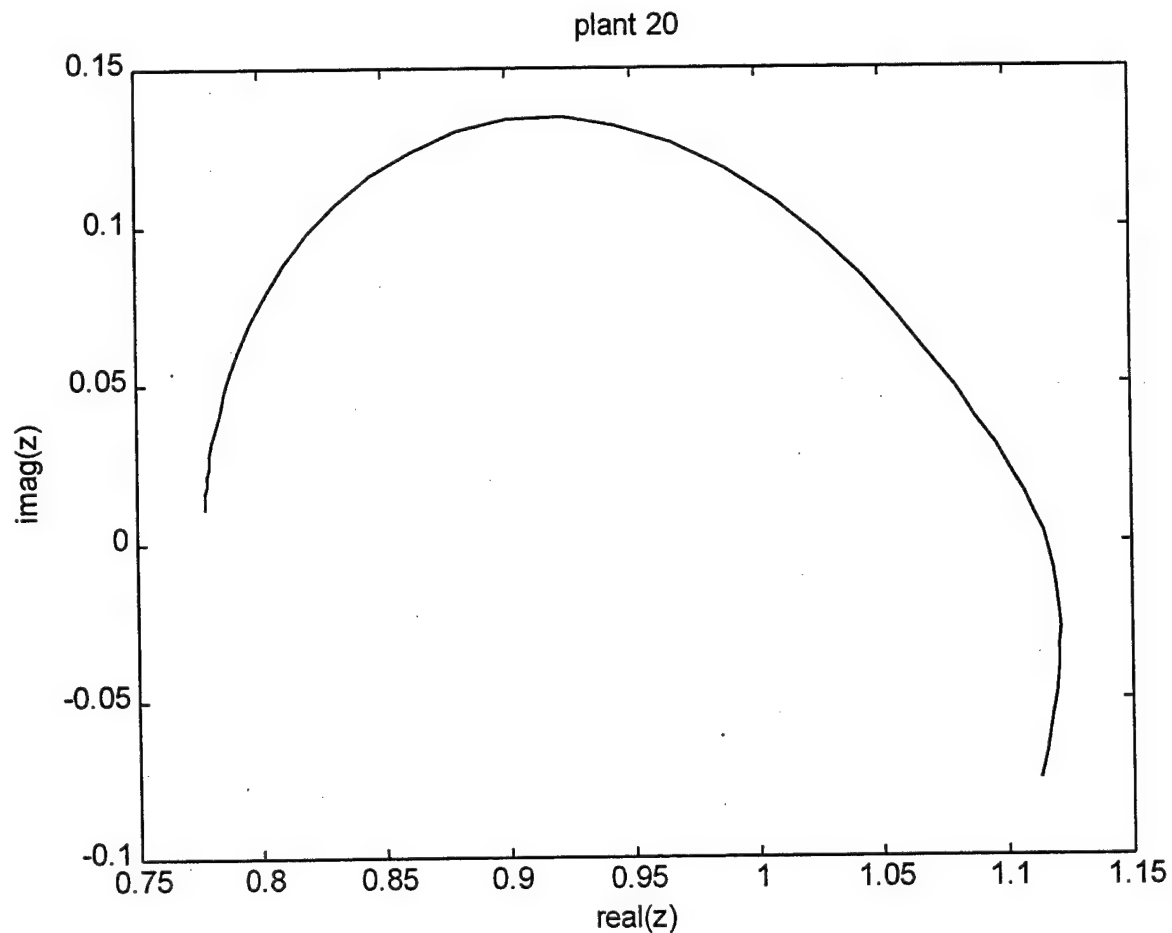


**Figure 5.2 Ratio of Determinants for a Path Connected Plant**
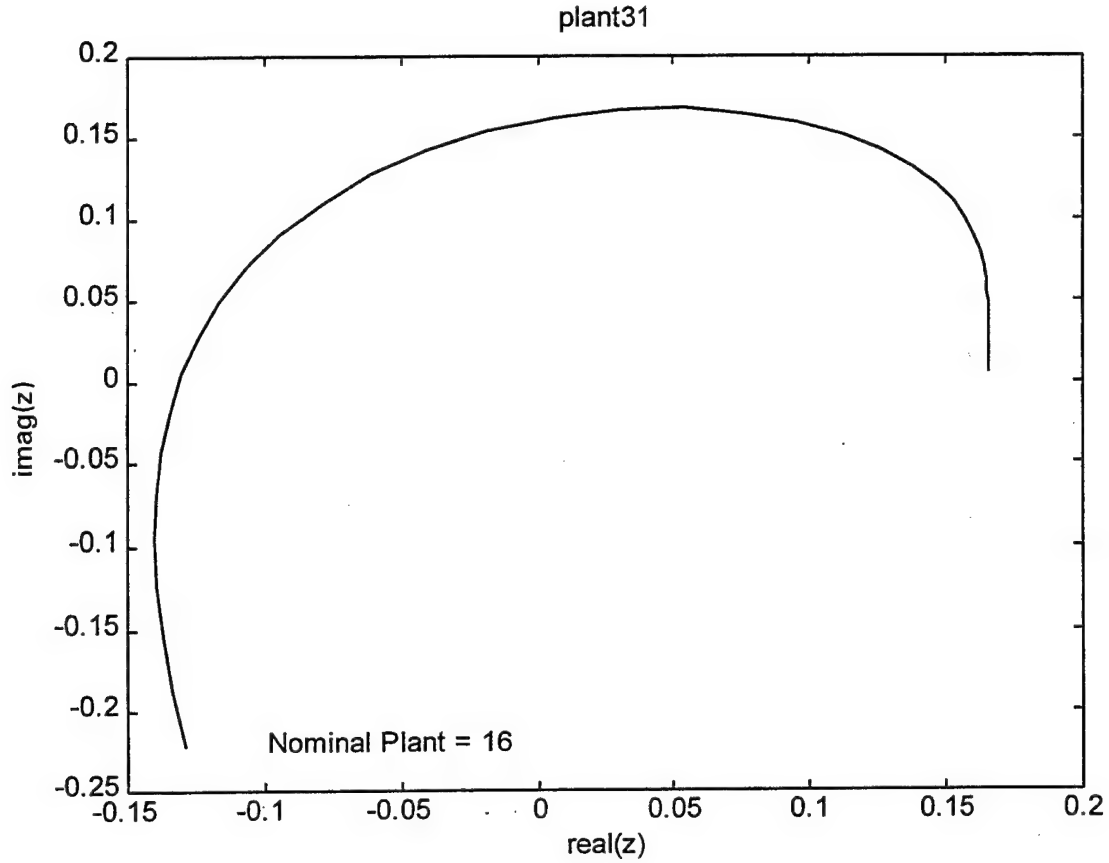
**Figure 5.3 Ratio of Determinants for a Nonpath Connected Plant**

To further improve the overall design the acceptable plant set is reduced by investigating the open loop plant responses when perturbed using a unit step. Figure 5.4 shows the Thrust and Stall margin responses to individual inputs of fuel flow and nozzle area. Note that for the thrust response to fuel flow input that characteristics tend to group into two groups, one with a fast time response and the other with a slower response. This characteristic continues for the stall margin response to a change in nozzle area. However, it is not as pronounced. This means that there is more uncertainty in the response to the input. Because of this the thrust response is used to identify the subset of acceptable plants to use and the stall margin response will be worked with. This choice

could be different if engine performance was not of primary interest. Figure 5.5 shows the open loop responses for the acceptable subset of plants. These are the plants that the subsequent design was done on. Note that the additional plants are not forgotten. After the design is complete these plants will be evaluated to determine how well they meet the specification. Then if need be, the specification can be changed or a new design can be performed for this set of plants. This is the traditional engineering problem.
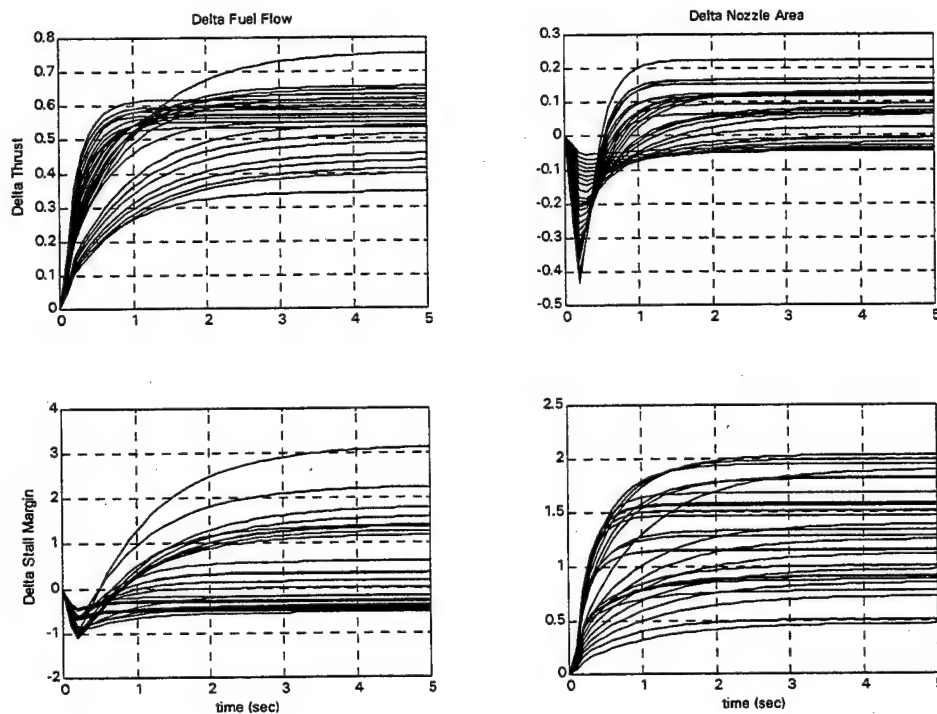


**Figure 5.4 Open Loop Responses for the Family of 29 Plants**
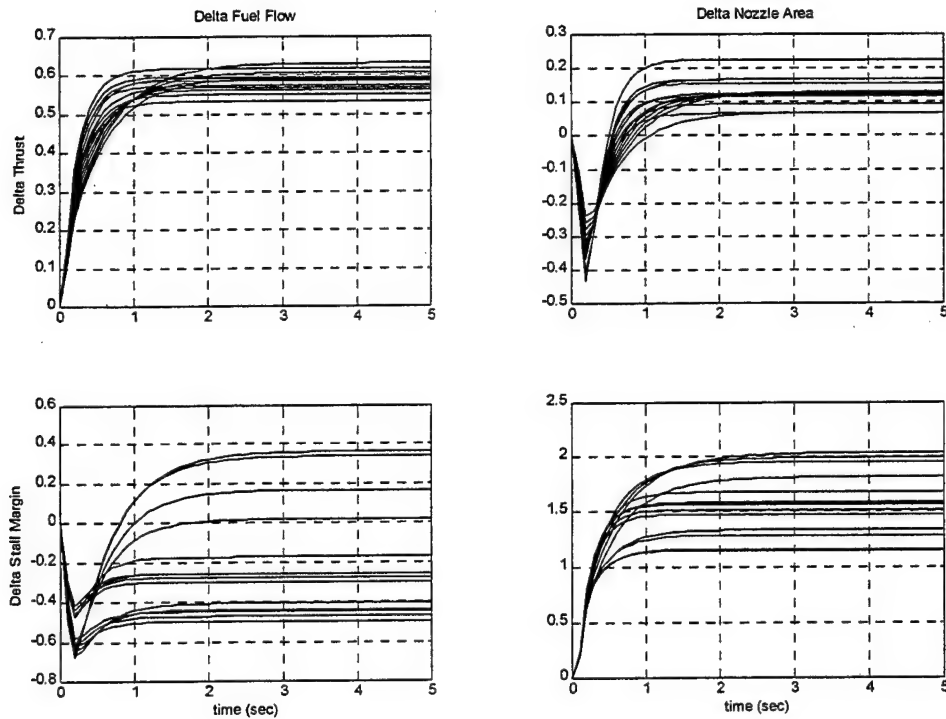
**Figure 5.5 Open Loop Responses for the Family of 14 Plants**

## 5.2    Control Objective and Specifications

The control objective is to regulate the two engine outputs (Thrust and Stall margin) that are unmeasurable using the two independent variables (Fuel Flow and Nozzle area) to achieve some overall response specification.  For this problem there are different response characteristics for each loop.  A time domain envelope described by both an upper and lower second order response models provides the regions of acceptable design.  These models are given by:

$$y_{1u}(s) = \max\left\{\frac{64}{s^2+9.6s+64}\cdot\frac{1}{s}, \frac{9}{s^2+3.6s+9}\cdot\frac{1}{s}\right\}$$

$$y_{1\ell}(s) = \min\left\{\frac{169}{s^2+31.2s+169}\cdot\frac{1}{s}, \frac{36}{s^2+14.4+36}\cdot\frac{1}{s}\right\}$$

[5.1]

$$y_{2u}(s) = \left\{\frac{36}{s^2+24s+36}\cdot\frac{1}{s}\right\}$$

$$y_{2\ell}(s) = \left\{\frac{16}{s^2+16s+16}\cdot\frac{1}{s}\right\}$$

A nominal plant model was chosen arbitrarily to be plant number sixteen out of the family of twenty-nine plants. This corresponds to the engine being at sea level static and a percent corrected rotor speed of 85%. The normalized nominal models, without the computational delays, actuator and sensor dynamics and transport delays is given in state-space form for the control and tracking filter designs in equations [5.2] and [5.3] respectively. These equations differ only by the output equations.

$$\begin{bmatrix} \dot{N} \\ \dot{P}_4 \\ \dot{P}_6 \end{bmatrix} = \begin{bmatrix} -4.5999 & 6.5570 & -11.6598 \\ 216.7323 & 20 & 214.0861 \\ 11.8577 & 113.0807 & -557.0501 \end{bmatrix} \begin{bmatrix} N \\ P_4 \\ P_6 \end{bmatrix} + \begin{bmatrix} 2.0399 & 0.0000 \\ 168.8145 & 0.0000 \\ 9.1891 & -245.9792 \end{bmatrix} \begin{bmatrix} w_f \\ A_8 \end{bmatrix}$$

$$\begin{bmatrix} Fn \\ SM \end{bmatrix} = \begin{bmatrix} 0.0305 & -0.0127 & 1.1570 \\ 1.9064 & -1.8457 & 0.0000 \end{bmatrix} \begin{bmatrix} N \\ P_4 \\ P_6 \end{bmatrix} + \begin{bmatrix} 0.0235 & 0.2312 \\ 0.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} w_f \\ A_8 \end{bmatrix}$$

[5.2]

$$\begin{bmatrix} N \\ P_3 \\ P_5 \end{bmatrix} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0400 & 0.0000 \\ 0.0000 & 0.0000 & 1.0200 \end{bmatrix} \begin{bmatrix} N \\ P_4 \\ P_6 \end{bmatrix} + \begin{bmatrix} 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \\ 0.0000 & 0.0000 \end{bmatrix} \begin{bmatrix} w_f \\ A_8 \end{bmatrix}$$

[5.3]

## 5.3 Tracking Filter Design

The tracking filter was designed using the methodology described in Chapter 3. A tracking filter was sought that converged the model outputs to the plant outputs quickly while at the same time providing good thrust and stall margin computations. Unfortunately, there exists no specification for the tracking filter response. Therefore, a specification was established that the eigenvalue (or pole locations) should be located so that a second order response achieved steady-state approximately twice as fast as the fastest responding plant within the plant set. This implies that the design should be done with the plant that has the slowest response. While all the plants had approximately the same transient response, it was noticed that the steady-state error was greater at lower speeds. Therefore, a lower speed condition was selected in order to guarantee that the proportional plus integral observer offered excellent results at all conditions. Plots of engine and engine model outputs before tracking, are shown in Figure 5.6.

After the successful design of the Proportional plus Integral (PI) Observer the PI elements for each loop were found to be.

$$
\begin{aligned}
PI1 &= \frac{50s + 50}{s} \\
PI2 &= \frac{27s + 40}{s} \\
PI3 &= \frac{31s + 45}{s}
\end{aligned}
\qquad [5.4]
$$

In Figure 5.7 a comparison of the outputs is given with the tracking filter switched on. These results indicate that the computation of thrust and stall margin will be good at all conditions.
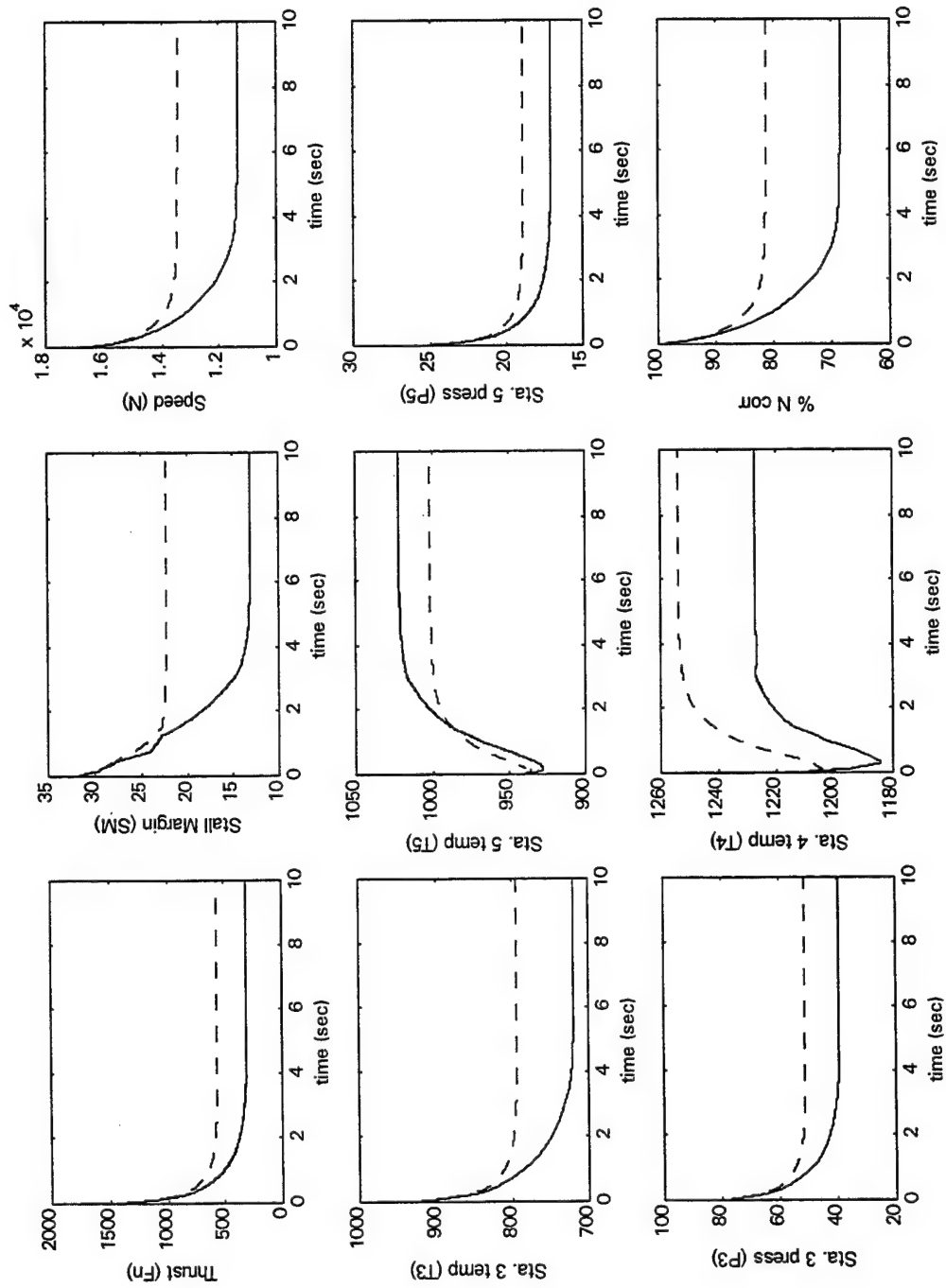
**Figure 5.6 Outputs Given the Tracking Filter is Off (- engine, -- model)**
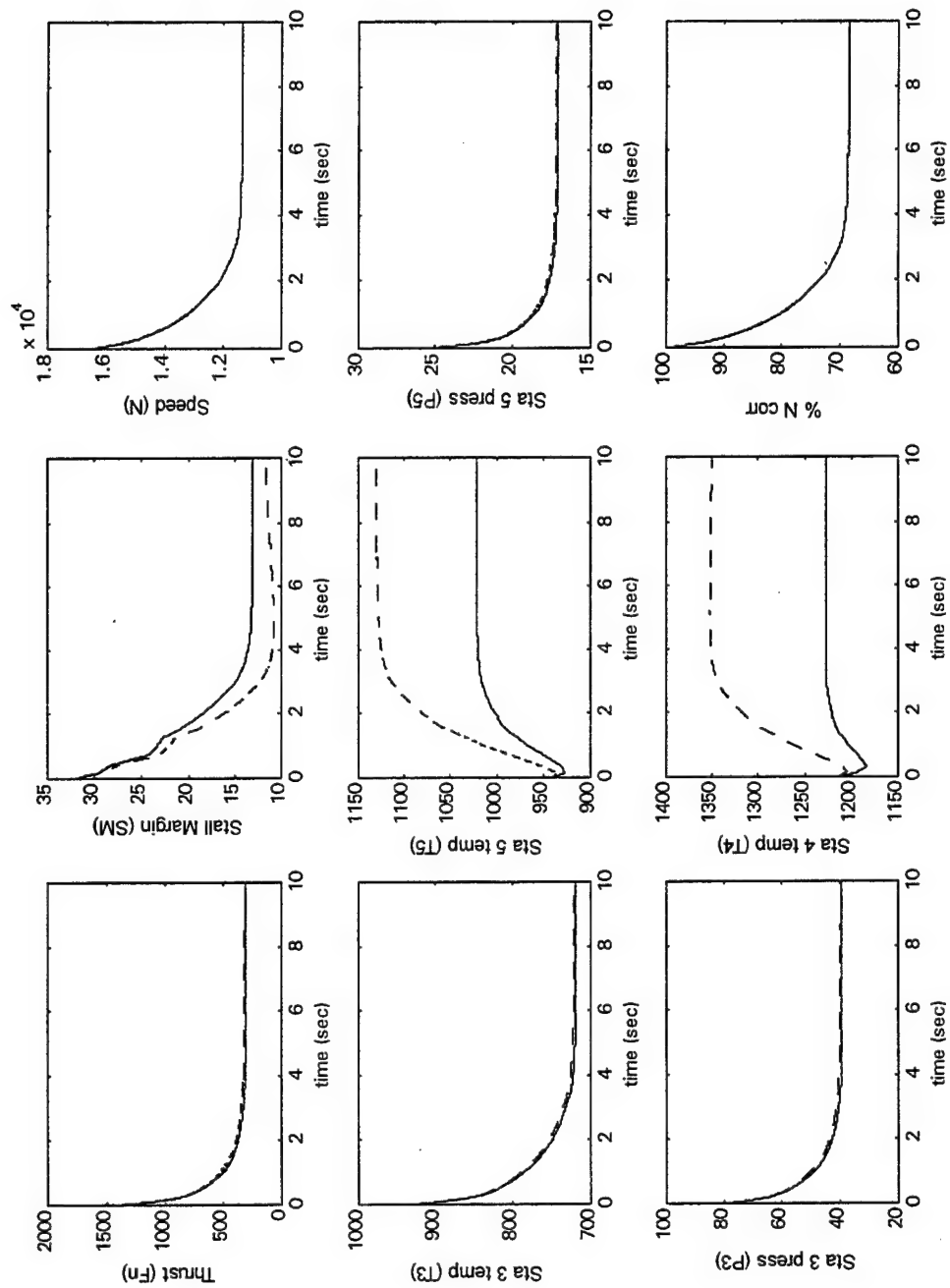
**Figure 5.7 Outputs Given the Tracking Filter is On (– engine, -- model)**

66

An important fact to note is that all outputs were not required to be tracked with zero steady-state error. In fact some parameters will not track in order to guarantee tracking of the outputs of interest. This has significant implications if the model is used for purposes other than achieving engine performance. Such an example is engine health monitoring where the tracking of component life is important. In this case accurate knowledge of the temperature levels that the components have been exposed to have a dramatic impact on the life remaining in the component. Therefore, temperatures would not be allowed to have as large an error as shown in Figure 5.7.

In Figure 5.8 a step in the demanded fuel flow was introduced to determine how well the model would then track the plant after the steady state has been achieved. This plot demonstrates that once the system is matched then it is consistent given a disturbance or movement to another operating condition.

## 5.4    Forward Loop Design

After the tracking filter has been designed, the control design can begin. Control system design becomes somewhat trivial when all parameters are available. However, in this example where thrust and stall margin are being controlled simultaneously there is significant amounts of interaction between the inputs and the outputs. Thereby complicating the design. The control structure of Chapter 4 is chosen, which includes a decoupling precompensating system and a diagonal closed loop controller. Finally, a prefilter is necessary to shape the output responses such that the time domain response specifications are satisfied to the degree feasible. In addition, the dynamics or time delay associated with the updating of the model, and the time required for the model to provide an output for feedback must be accounted for. Initial investigations indicate that due to

67
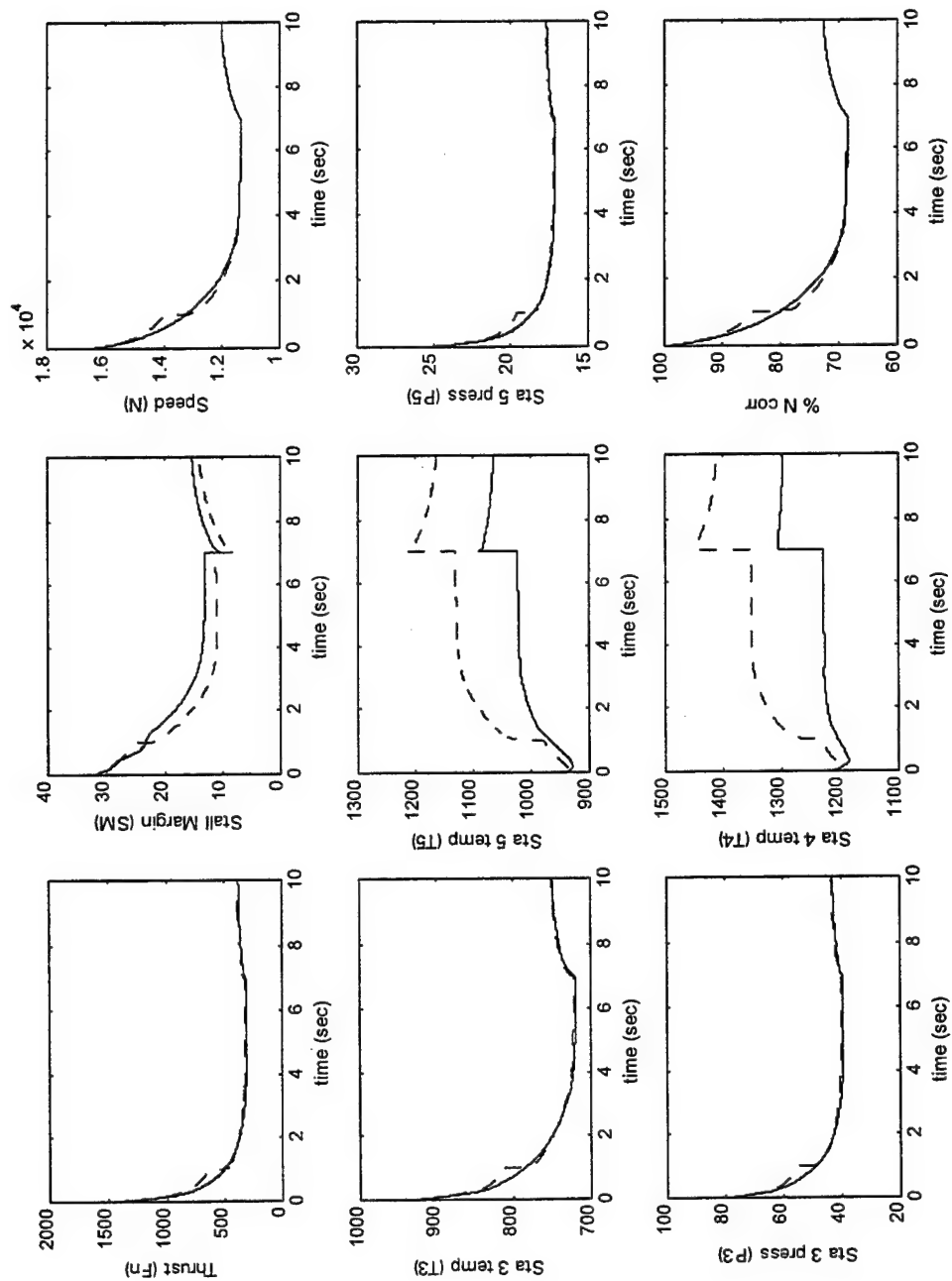
Figure 5.8 Outputs Given the Tracking Filter is On with A Step in One Input (- engine, -- model)

time delays in the sensors and actuators the maximum achievable loop gain cross-over frequencies, over all the loops and over the uncertainty range of the plant, and at which we expect the interaction will be critical, will be in the range of 1.0 rad/s to 6.0 rad/s.

**5.4.1 Precompensator Design.** The role of the precompensating system is to reduce the interaction index of the plant as much as possible. The interaction index for the plant is given in Figure 5.9. At low frequencies the interaction index is nearly less than a third indicating that the system is almost decoupled. However, for the bandwidth of concern some of the plants have interaction index greater than one, therefore, stability cannot be guaranteed for these plants.
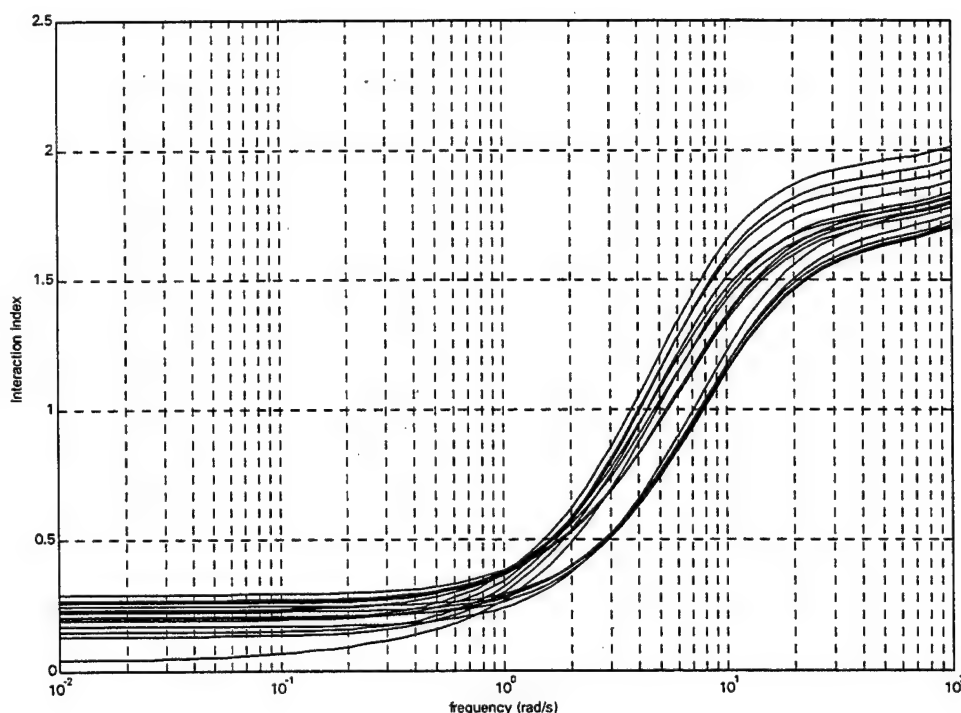


**Figure 5.9 Interaction Index for Open Loop Plants**

The precompensating system has unity on the diagonal and the elements to be designed on the off-diagonal. These elements were originally designed to be static compensators. But, due to a push-pop effect it was found that static compensators actually made the interaction index worse at low frequencies. This was unacceptable, therefore, a dynamic compensator was designed. The off-diagonal elements of the pre-compensator are given as:

$$k_{12} = \frac{0.05\left(\dfrac{s}{0.1034}+1\right)\left(\dfrac{s}{2.144}+1\right)}{\left(\dfrac{s}{1.547}+1\right)\left(\dfrac{s}{29.86}+1\right)}$$

$$k_{21} = \frac{0.005\left(\dfrac{s}{0.02}+1\right)}{\left(\dfrac{s}{2.087}+1\right)}$$

[5.5]

Figure 5.10 shows the resulting plot of interaction index. Notice that for all frequencies within the bandwidth, the interaction index is less than one, thereby stability is guaranteed. Also, through the known system bandwidth the interaction index was approximately one-third or less, so that the almost decoupling condition has been achieved. This greatly simplifies the controller design covered in the next section.

**5.4.2 Diagonal Controller Design.** The methodology and equations presented in Chapter 4 were used as the basis for designing the diagonal controllers. This methodology results in the least conservative design in that they utilize all available parametric uncertainty information available. For this design, it did not matter which

70

loop you started with. However, loop one was selected due to the tighter behavior of the

step responses shown in Figure 5.5.
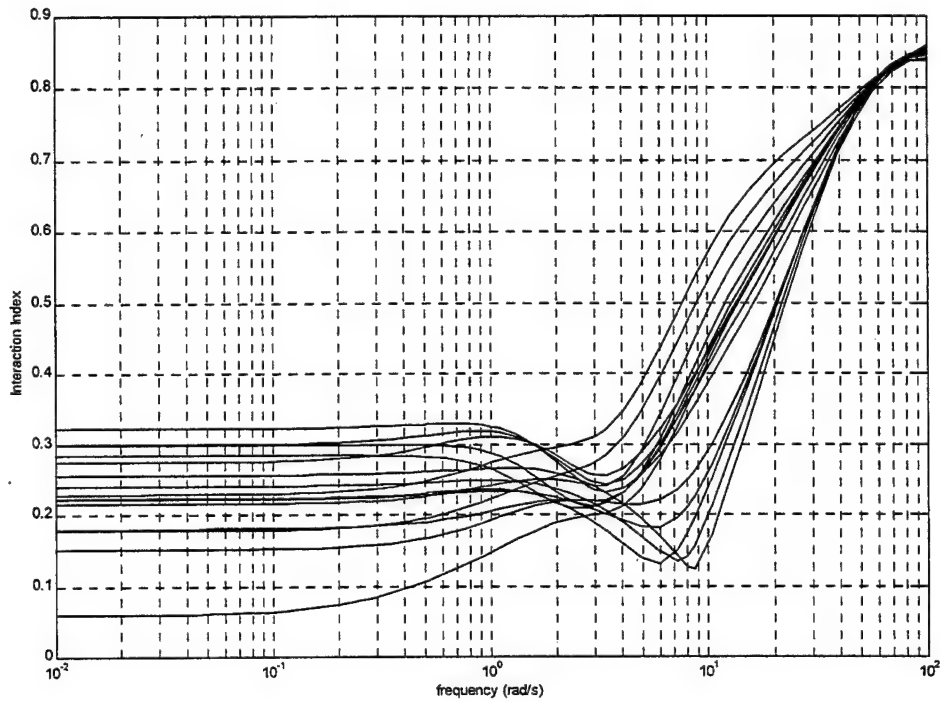


**Figure 5.10 Open Loop Interaction Index for Plant and Precompensator**

The controllers were determined to be:

$$g_{11} = \frac{7\left(\dfrac{s^2}{100} + \dfrac{15s}{100} + 1\right)\left(\dfrac{s}{4.201} + 1\right)\left(\dfrac{s}{4.958} + 1\right)}{s\left(\dfrac{s}{5} + 1\right)\left(\dfrac{s}{8.560} + 1\right)\left(\dfrac{s}{69.06} + 1\right)\left(\dfrac{s}{120} + 1\right)}$$

$$g_{22} = \frac{0.8\left(\dfrac{s}{0.7025} + 1\right)\left(\dfrac{s}{2.38} + 1\right)}{s\left(\dfrac{s}{0.5124} + 1\right)\left(\dfrac{s}{6.723} + 1\right)}$$

[5.6]

71

**5.4.3 Prefilter Design.** In order to achieve a faster cleaner response, a prefilter is placed at the input to each channel. The prefilter is designed to force the individual channel closed loop frequency responses to lie between the upper and lower step response bounds. Figures 5.11 and 5.12 show the frequency responses of the individual channels plotted against their respective upper and lower step response specifications. These prefilters are given as follows:

$$f_{11} = \frac{1}{\left(\dfrac{s}{3.963} + 1\right)}$$

[5.7]

$$f_{22} = \frac{\left(\dfrac{s}{2.703} + 1\right)\left(\dfrac{s}{11.96} + 1\right)}{\left(\dfrac{s}{1.32} + 1\right)\left(\dfrac{s}{3.139} + 1\right)}$$

## 5.5    Design Analysis

The resulting closed loop time domain step responses for the feedback system are shown in Figures 5.13 and 5.14. The responses are presented in columns where column 1 shows the two outputs for a step input on channel one. Column two shows the corresponding outputs for a step input on channel two. Figure 5.13 shows the output responses for the feedback system without the prefilter, and Figure 5.14 shows the corresponding outputs with the prefilter. The first channel shows a settling time on the order of two seconds, and the second channel settling time is on the order of five seconds. Channel one appears to have the most uncertainty. This is attributed to the significantly larger time delay present in this loop.

72
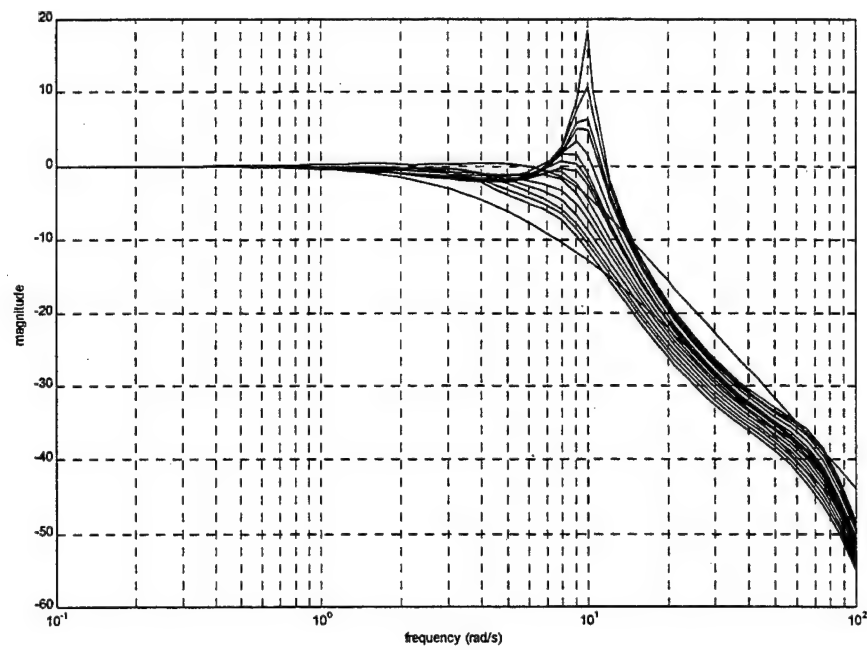
**Figure 5.11 Frequency Response for Thrust to Fuel Flow Channel**
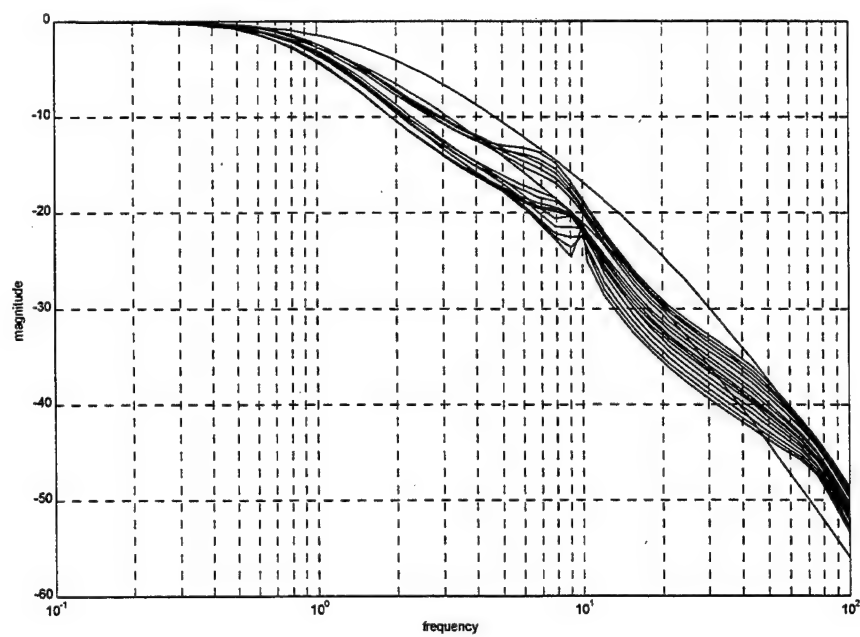


**Figure 5.12 Frequency Response for Stall Margin to Nozzle Area Channel**

73
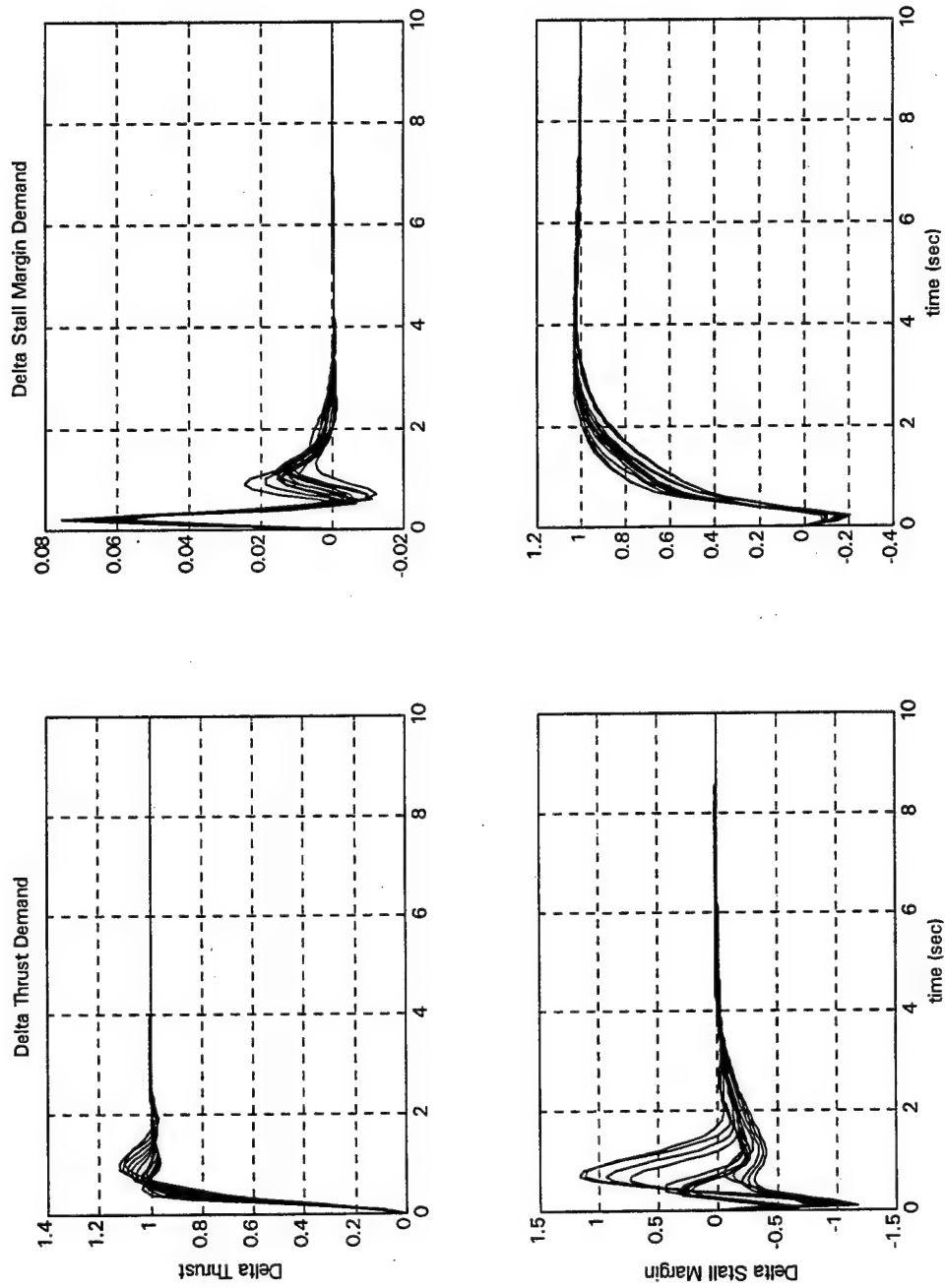
**Figure 5.13 Compensated Step Responses (No Prefilter).**

74
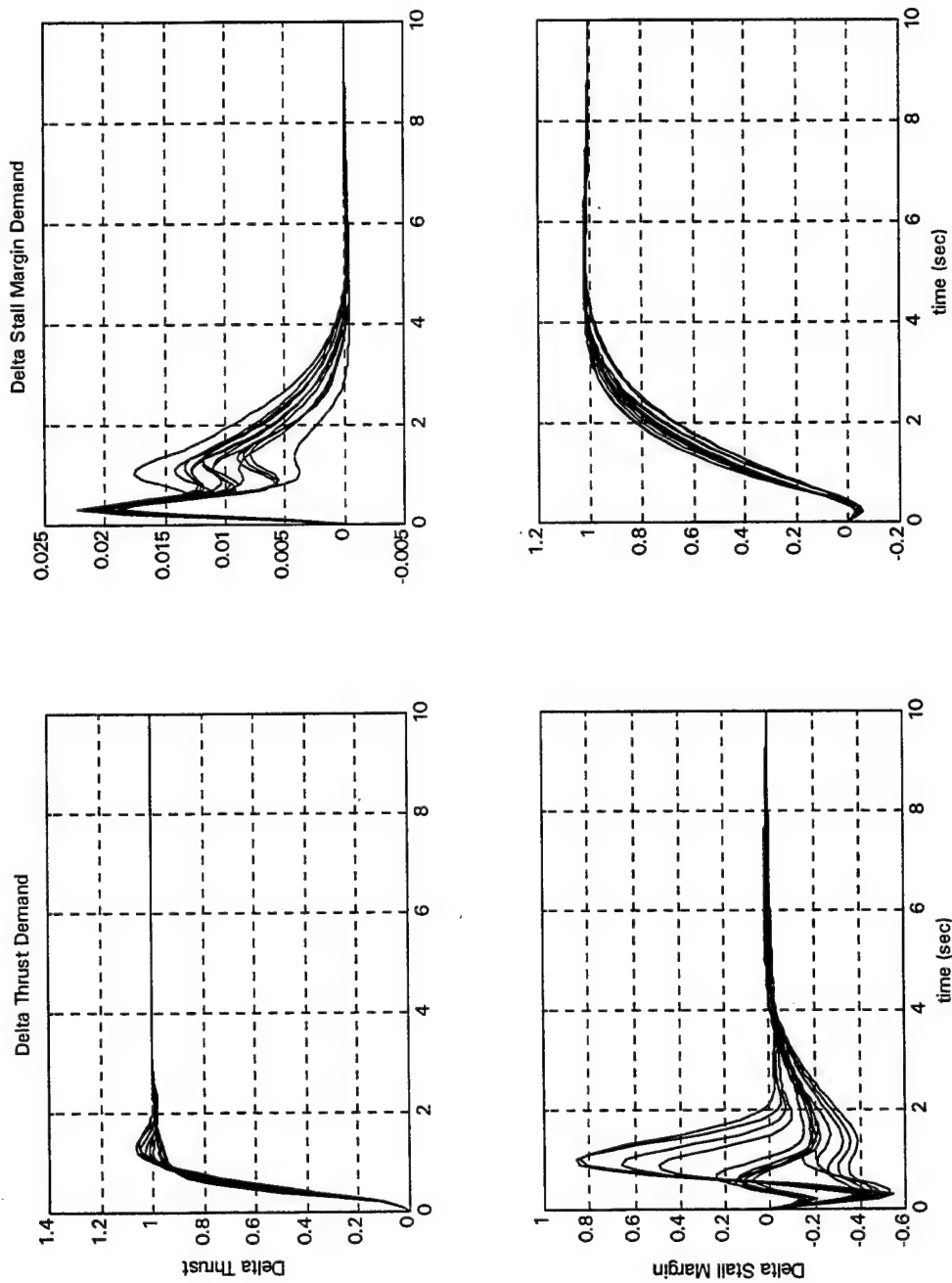
**Figure 5.14 Compensated Step Responses (with Prefilter).**

# CHAPTER 6

## CONTRIBUTIONS, CONCLUSIONS AND RECOMMENDATIONS

### 6.1    Summary of Contributions

The model-based control concept is a significant departure from the typical aircraft engine control system design. Until recently, the computing power available to the engine control designer has been one of the main limiting factors. By utilizing the improvements in today's computing equipment, the functionality of the engine control can be significantly increased. The model-based control concept capitalizes on this increase in computing power so that the designer can control the engine in ways never before possible. The work presented herein contributes in several ways to the overall knowledge of engine multivariable control system design. The ideas presented are shown in Chapter 5 to be very effective when applied to a turbojet design problem.

Model-based control (MBC) can be categorized as an adaptive control technique. However, model-based control differs from classical adaptive control, one of the main techniques is model-reference adaptive control (MRAC), in at least two key areas. The first difference is with the controllers. In an MRAC system the parameters of the controller are adjusted to obtain a desired response while in a MBC system the controller parameters are fixed. The second difference and probably the most significant difference is with the models. In the MRAC system the model represents the ideal plant response to the input. The model in the MBC system is a simulation of the plant. The adaptation in

the MBC system occurs in the simulation by adjusting the parameters of the simulation through an adaptation law (tracking filter) so that the outputs of the simulation match certain key outputs of the plant when subjected to the same inputs. The simulation can then compute the desired feedback values, once the error between the outputs of the plant and simulation is below some threshold.

In addition, the idea of robustness became paramount when working with the turbine engine and developing a control design philosophy. The engine behaves or responds differently not only at the many flight conditions at which it will operate, but also the characteristics will be different based upon the throttle setting at a given flight condition. Capturing of these differences and including them in a single design methodology is a very challenging task. In every step of the design process, both structured and unstructured uncertainty is considered to ensure robustness of the design.

The introduction of Quantitative Feedback Theory (QFT) as a design tool in the model-based control structure is a significant benefit to the control system designer. QFT allows the establishment of constraints in an orderly and concise manner so that a low-order, robust controller is found that satisfies overall system specifications. Also, the development presented in this work establishes a solid mathematical foundation for the design of robust model-based control system.

Another key area of contribution, which is covered in Chapter 3, is the development of a robust tracking filter. The tracking filter is the heart of any model-based control system. If the model can not be adjusted so that it is an accurate representation of the plant being controlled, then the variables computed by the model, in particular the ones that will ultimately be used for feedback, will not be accurate enough.

In this work, a method for updating the states of the model using proportional-plus-integral observer theory is presented. This concept was then extended to the multivariable case.

The next area of contribution, presented in Chapter 4, was a further refinement of the use of the Perron root of the interaction matrix as a measure of the generalized diagonal dominance of uncertain multivariable plants. This capitalizes on non-negative matrix theory and the use of H- and M-matrices. Earlier works present a method for designing a feedback loop around the plant in order to achieve a certain level of decoupling (Nwokah, et.al., 1995). However, in this work a methodology for designing a dynamic forward loop precompensator is developed. A static precompensator was initially considered for this design. However, the interaction index of the plant was very sensitive to the design of a static precompensator. This was exhibited by a push-up in the interaction index at low frequencies, which is undesirable. By designing a dynamic compensator the engineer now has the ability to shape the interaction index curve, thereby allowing the interaction index to stay small at low frequencies. The design of the precompensator is not a standard part of the QFT design process. But, for any practical controller design the engineer desires to have a system that is as decoupled as possible. The inclusion of this block permits the designer to attempt to achieve this desirable characteristic. Then, using the Quantitative Feedback Theory the overall control design process is greatly simplified.

Finally, in Chapter 5, the above theoretical and practical developments were applied to an engine control problem. This problem is unique in that the feedback variables are not directly sensed. They are computed within the embedded model and

78

then used to compute error signals for the controller. In order to include additional realism into the problem, transport delay and combustion delays were considered in the model of the engine. Time delay was also considered for the execution of the control law and for the sensors and actuators. The inclusion of this additional time delay made the design work much more difficult to accomplish. However, the controller demonstrated exceptional time and frequency domain behaviors, while requiring minimal gain over the control bandwidth.

## 6.2    Conclusions

A number of important lessons were learned during the execution of the design. The most significant of these concerned the limitation of achievable system performance in the presence of system time delay. This time delay caused many difficulties while trying to shape the loops to achieve overall system level specifications. Methods for dealing with non-minimum phase systems are known. However, it appears that accounting for this behavior and its effects becomes a unique aspect of every design.

Another key lesson was the push-pop effect that occurs when designing static precompensators. Even though this phenomenon was noticed in the design structure presented by Nordgren (1994), it appears to be more pronounced problem when the decoupling system is in the forward path. This effect made it necessary to design a dynamic compensator to keep the interaction between the control loops reasonable. This design was relatively simple and offered a lot of benefit to the overall process.

To the best of our knowledge, there exists no robust, model-based, multivariable control design methodology for the engine control problem. Further use of this concept

79

and refinement of the methodologies will greatly improve the overall engine system. This could especially be true in other areas related to control such as engine health monitoring.

## 6.3    Recommendations

There are many additional topics or areas of improvement that became apparent during the performance of the work presented herein. The first concerns the modeling of the engine. During an investigation of existing engine modeling techniques, different approaches for modeling the dynamics of the system were identified. Some of the models were iterative and would converge to a steady state answer before returning values. While others accepted small errors during the transient and would provide values through each pass of the code. While each process will work, careful consideration must be given to the time delay and overall impact of carrying small errors through a transient will have on the model-based control concept. Better methods of representing the components may offer a significant advantage of current representations. Typical engine models use lookup tables to represent the components; this is the method used for the nonlinear engine model in this work. However, to improve the functionality of the model a neural net could be used to represent the components thereby possibly capturing additional component characteristics.

The second concerns the expansion of the design to include the remainder of the flight envelope. This will not be a major concern as long as plant response characteristics are similar and the plant models are path connected. However, if this is not the case then multiple controller designs are required to cover the entire envelope. If multiple designs

are required then a methodology for connecting or translating between the different controller designs must be used. One approach presented by Polly, et. al., (1988), uses a curve fitting approach to connect all the gains of the controllers together. It is not known if this approach will work well with this design technique or if another approach must be found.

While the tracking filter appears to be such a simple element in the overall concept, the importance of this element can not be over emphasized. To date, only tracking filters that work well for a single mode at a time have been found. The desire is to find a multipurpose tracking filter, a filter that works well for all modes of operation. The basic limitation is the number of sensed values from the plant that can be used to update parameters in the model. Research needs to continue in order to find a tracking filter that meets a multitude of operational needs or identifies a satisfactory approach to having multiple tracking filters.

The final area of interest concerns mode selection. In this work a single mode (performance) was shown. One advantage of model-based control is that the engine can have more than one mode of operation. Consider a fighter aircraft. During the lifetime of the aircraft it will be used in many different roles such as combat, training and escort missions. Each of these roles uses and requires different capabilities from the same engine hardware. For combat, performance is of primary concern while for training, the protection and safety of the hardware is more important and for the escort mission, fuel burn is of interest. Also, the pilot could switch between the different modes, depending upon the current mission need. The use of the model-based control concept will bring new and powerful capabilities to the engine system designer and user in the future.

81

# REFERENCES

Adibhatla, S., H. Brown, and Z. Gastineau (1992). Intelligent Engine Control, AIAA-92-3484. Presented at the 28th Joint Propulsion Conference. Nashville, Tennessee.

Adibhatla, S. and Z. Gastineau (1994). Tracking Filter Selection and Control Mode Selection for Model Based Control, AIAA-94-3204. Presented at the 30th Joint Propulsion Conference. Indianapolis, Indiana.

Araki M. and O.D.I. Nwokah (1975). Bounds for closed-loop transfer functions of multivariable systems. IEEE Transactions on Automatic Control, 17:666-670.

Athans, M., P. Kapasouris, E. Kappos, and H.A. Spang (1985). Linear-quadratic gaussian with loop-transfer recovery methodology for the F-100 engine. AIAA Journal of Guidance, Control and Dynamics, 9(1): 45-52.

Bailey, F.N., J.W. Helton and O. Merino (1994). Alternative approaches in frequency domain design of single loop feedback systems with plant uncertainty. In Proceedings of the 1994 American Control Conference, Baltimore, MD.

Beale, S. and B. Shafai (1989). Robust control system design with a proportional integral observer, International Journal of Control, vol. 50, no. 1, 97-111.

Berman, A. and R. J. Plemmons (1979). Non-Negative Matrices in The Mathematical Sciences, Academic Press Inc., New York.

Brown, S. (1990). 21st Century Hot Jet Engines. Popular Science, June 1990. pg. 83-89.

Bode, H. (1945). Network Analysis and Feedback Amplifier Design. Van Nostrand, New York.

Chaing, R.Y. and M.G. Safonov (1988). MATLAB Robust Control Toolbox. The Math Works, Inc.

Doyle, J.C. and G. Stein (1979). Robustness with observers. IEEE Transactions on Automatic Control, 7:1-17.

Doyle, J.C. and G. Stein (1981). Multivariable feedvack design: concepts for a classical / modern synthesis. IEEE Transactions on Automatic Control, 26:4-16.

Doyle, J.C., A.R. Tannenbaum, and B.A. Francis (1992). Feedback Control Theory. McMillan, New York.

Francis, B.A. (1987). A course in H∞ Control Theory. Springer Verlag, New York.

Freudenburg J.S. and D.P. Looze (1987). Frequency Domain Properties of Scalar and Multivariable Feedback Systems. Springer Verlag, New York.

Gastineau, Z.D., G. Happawana and O.D.I. Nwokah (1998). Robust Model-Based Control for Jet Engines, AIAA-98-3752. To be presented at the 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference.

Gastineau, Z.D., G. Happawana, and O.D.I. Nwokah (1998). Model-Based Control for Jet Engines. Submitted to the AIAA Journal of Dynamics and Control

Geyser, L. C. (1979). DYGABCD – a program for calculating linear A, B, C, and D matrices from a nonlinear engine simulation. Technical Paper NASA TP 1295, NASA – Lewis Research Center.

Horowitz, I.M. (1963). Synthesis of Feedback Systems. Academic Press.

Horowitz, I.M. and M. Sidi. (1972). Synthesis of feedback systems with large plant ignorance for prescribed time-domain tolerances. International Journal of Control, 16(6):287-309.

Horowitz, I.M. and M. Sidi. (1978). Optimum synthesis of non-minimum phase feedback systems with plant uncertainty. International Journal of Control, 27(3):361-386.

Horowitz, I.M. (1979). QFT for uncertain MIMO systems, International Journal of Control, vol. 30, pp81-106.

Horowitz, I.M. (1991). Survey of quantitative feedback theory. International Journal of Control, 53:255-291.

Jayasuriya, S., and Y. Zhao (1992). Stability of quantitative feedback designs and the existence of robust QFT controllers. In C. H. Houpis and P.R. Chandler, editors, Quantitative Feedback Theory Symposium, pages 503-541, Wright-Patterson Air Force Base, Ohio. Also in International Journal of Robust and Non-linear Control, (1993).

Kaczorek, T. (1979). Regelungstechnik, 27, 359-362.

Kailath, T. (1980). Linear Systems, Prentice Hall, Englewood Cliffs, NJ

Kapasouris, P., M. Athans, and A. Spang (1985). Gain-scheduled multivariable control for the GE-21 turbofan engine using the LQG/LTR methodology, In Proceedings of the 1985 American Control Conference, pages 109-118

Kharitonov, V.L. (1979). Asymptotic stability of an equilibrium position of a family of linear differential equations, Differential Equations, 14:1483-1485.

Leininger, G.G. (1981). Multivariable nyquist array method with application to turbofan engine control. Technical Paper NASA-CP-2137, NASA-Purdue University.

Maciejowski, J.M. (1989). Multivariable Feedback Design, Addison Wesley.

Mattingly, J. D., W. H. Heiser, and D. H. Daley (1987). Aircraft Engine Design, AIAA Press, Washington, D.C.

Misawa, E. A. and J. K. Hedrick (1989). Nonlinear Observers: A state-of-the-art survey. ASME Journal of Dynamic Systems, Measurement and Control. Volume 111, pp 344-352.

Merrill, W., W., B. Lehtinen, and J. Zeller (1984). The role of modern control theory in the design of controls for aircraft turbine engines. AIAA Journal of Guidance and Control, 7(6):652-661.

Nordgren, R.E., O.D.I. Nwokah, and M.A. Fanchek (1994,1993). New formulations for quantitative feedback theory. International Journal of Robust and Nonlinear Control, 4:47-64. Also in Proceedings of the 1993 American Control conference, San Francisco, CA, pages 1716-1720.

Nordgren, R. E., Z. Gastineau., S. Adibhatla, G. Grewal, and O. D. I. Nwokah (1994). Robust Multivariable Turbofan Engine Control: A Case Study, Proc. IEEE Conference on Decision and Control, Lake Buena Vista, Florida

Nwokah, O.D.I. (1984). Synthesis of controllers for uncertain multivariable plants for prescribed time domain tolerances. International Journal of Control, 40:1189-1206.

Nwokah, O.D.I. (1988). Strong robustness in uncertain multivariable systems. In IEEE Conference on Decision and Control, Austin, TX.

Nwokah, O.D.I. and D. F. Thompson (1989). Algebraic and topological aspects of quantitative feedvack theory. International Journal of Control, 50:1057-1069.

Nwokah, O.D.I., S. Jayasuriya, and Y. Chait (1992). Parametirc robust control by quantitative feedback theory. AIAA Journal of Guidance, Control and Dynamics, 15(1):207-214.

Nwokah, O.D.I., R.E. Nordgren and G.S. Grewal (1995). The inverse nyquist array: A quantitative theory. IEE Proc. –Control Theory Appl, Vol. 142, No. 1, January 1995, pp23-30.

Nwokah, O.D.I., R.E. Nordgren, and G.S. Grewal (1994). Optimal loop transmission functions in SISO quantitative feedback theory. In Proceedings of the 1994 American Control Conference, Baltimore, MD.

Oates, G. C. (1988). Aerothermodynamics of Gas Turbine and Rocket Propulsion, AIAA Press, Washington, DC.

Perez, R. A. and O. D. I. Nwokah (1991). Full envelope multivariable control of a gas turbine engine. In Proceedings of the 1991 American Control conference, pages 735-740, Boston, MA.

Perez, R. A. and O. D. I. Nwokah (1991). Model reference control of a linear plant with feedthrough element. In Proceedings of the IEEE International Conference on Systems Engineering, pages 81-84.

Petty, J. and R. Henderson (1987). The coming revolution in turbine engine technology. Presented at AGARD/PEP Symposium on "Advanced Technology for Aero Gas Turbine Components."

Pfeil, W. H., M. Athans and H. A. Spang (1986). Muti-variable control of the GE T700 engine using the LQG/LTR design methodology. In Prodeedings of the 1986 American Control Conference, Pages 1297-1309, Seattle, WA.

Polley, J.A., S. Adibhatla, and P.J. Hoffman (1988). Multivariable turbofan engine control for full flight envelope operation, 88-gt-6. In Gas Turbine and Aeroengine Congress and Exposition, Amsterdam, June 5-9 1988. Also in Transactions of the ASME, Journal for Power.

Qi, O. and N.R.L. Maccallum (1993). A Model-based Approach to the control of an aircraft gas turbine engine, 93-GT-402. Presented at the International Gas Turbine and Aeroengine Congress and Exposition. Cincinnati, Ohio.

Rosenbrock, H.H. (1969). Design of multivariable control systems using the inverse nyquist array. Proceedings of the IEE, 116(11).

Rosenbrock, H.H. (1970). State-Space and Multivariable Theory. Wiley, New York.

Sellers, J. F. and C. J. Daniele (1975). DYNGEN – a program for calculating steady-state and transient performance of turbojet and turbofan engines. Technical Note NASA TN D-7901, NASA-Lewis Research Center.

Shafai, B. (1985). Proceedings of the 24th Conference on Decision and Control, Fort Lauderdale, Florida, pp. 597-599.

Skertic, R. J. (1992). GE16 component level model and control model: version 1.0. Technical Report, General Electric company.

Skira,C. and M. Agnello (1991). Control Systems for the next Century's Fighter Engines, 91-GT-278. Presented at the International Gas Turbine and Aeroengine Congress and Exposition. Orlando, Florida.

Suh, N. P. (1990). Principles of Design, Oxford University Press,

Weinberg, M. S. (1975). A multivariable control for the F100 engine operating at sea level static. Technical Report ASD-TR-75-28, Wright-Patterson Air Force Base, OH.

Wojciechowski, B. (1978). Analysis and synthesis of proportional-integral observers for single-input single-output time-invariant continuous systems. Ph.D. dissertation, Gliwice, Poland.

Yau, C. H. and O. D. I. Nwokah (1994). Almost decoupling of uncertain multivariable systems, International Journal of Control, 58(6):1385-1408.

Yau, C. H., J.E. Gallagher, and O.D.I. Nwokah (1994). A model reference quantitative feedback design theory with application to turbomachinery. International Journal of Robust and Nonlinear Control, 4:181-210.

Zames, G. (1981). Feedback and optimal sensitivity: Model reference transformations, multiplicative seminorms, and approximate inverses. IEEE Transactions on Automatic Control, 26(2):301-320.

# APPENDICES

# APPENDIX A

## EQUIVALENCE RELATION DERIVATION

Consider a linear continuous time-invariant system described by the state-space equations

$$\dot{x} = Ax + Bu \qquad [A.1]$$

$$y = Cx + Du \qquad [A.2]$$

where x is an nx1 vector of state-variables, u is an mx1 vector of input functions, and y is an lx1 vector of outputs.

Solving equation [A.1] with the initial condition vector x(0), and substituting into equation [A.2], we obtain

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau + Du(\tau) \qquad [A.3]$$

From [A.3], x(0) can be determined given the vectors y(t) and u(t) over an interval if, and only if, the columns of the matrix $Ce^{At}$ are linearly independent. Hence, a necessary and sufficient condition for the system to be observable is that the rank of $\phi_0 = n$, where $\phi_0$ is the matrix

$$\phi_0 = \left[ C^T, A^T C^T, \left(A^T\right)^2 C^T, \ldots, \left(A^T\right)^n C^T \right] \qquad [A.4]$$

88

## A.1 Observability

The system described by [A.1] and [A.2] is said to be observable if there exists a time $t_1 > 0$ such that given the vectors u and y over the interval $(0, t_1)$ it is possible to deduce the initial state-vector $x(0)$.

If $\phi_0$ has rank n then the system is completely observable. If $\phi_0$ does not have rank n, then some of the system states are unobservable and have no influence on the system outputs. The rank deficiency of $\phi_0$ indicates the number of unobservable states but does not identify these.

## A.2 Observable Canonical Forms

If a system [A, B, C] is observable, then the matrix $\phi_0$ defined in equation [A.4] has rank n. A nonsingular nxn transformation matrix T can then be formed from the columns of $\phi_0$ as follows:

(a) Inspect the columns of $\phi_0$ from the left to the right and retain only those vectors which are linearly independent of those previously selected.

(b) Arrange the n linearly independent vectors so selected to form a new matrix ($\Gamma$), where

$$\Gamma = \left( c_1^t, A^t c_1^t, \ldots, \left(A^t\right)^{\mu_1 - 1} c_1^t, c_2^t, A^t c_2^t, \ldots, \left(A^t\right)^{\mu_\ell - 1} c_\ell^t \right) \qquad [A.5]$$

The integers $\left(\mu_1, \ldots, \mu_\ell\right)$ are known as the observability indices of the system, and

$$\sum_{i=1}^{\ell} \mu_i = n.$$

89

The parameter

$$\upsilon_o = \max_i \mu_i; \quad i = 1, \ell \qquad \text{[A.6]}$$

is often called the observability index for the system.

(c) Let $\Gamma^{-1}$ be described in terms of its rows as

$$\Gamma^{-1} = \begin{bmatrix} \gamma_1^t \\ \vdots \\ \gamma_n^t \end{bmatrix} \qquad \text{[A.7]}$$

and let $\gamma_{k_i}^t$ be the $k_i^{th}$ row of $\Gamma^{-1}$, where

$$k_i = \sum_{j=1}^{i} \mu_j; \quad i = 1, \ell \qquad \text{[A.8]}$$

(d) Using the vectors $\gamma_{k_i}^t$ the required transformation matrix T is formed as

$$T = \left[ \gamma_{k_1}, A\gamma_{k_1}, \ldots, A^{\mu_1 - 1}\gamma_{k_1}, \gamma_{k_2}, A\gamma_{k_2}, \ldots, A^{\mu_\ell - 1}\gamma_{k_\ell} \right] \qquad \text{[A.9]}$$

(e) The observable standard form for the system [A, B, C] is then given by

$$\tilde{A} = T^{-1}AT$$

$$\tilde{B} = T^{-1}B \qquad \text{[A.10]}$$

$$\tilde{C} = CT$$

where

90

$$\widetilde{A} = \begin{bmatrix} \widetilde{A}_{11} & \cdots & \widetilde{A}_{1\ell} \\ \widetilde{A}_{\ell 1} & \cdots & \widetilde{A}_{\ell\ell} \end{bmatrix} \qquad [A.11]$$

and the diagonal blocks $\widetilde{A}_{ii}$ are in the companion form

$$\widetilde{A}_{ii} = \begin{bmatrix} 0 & 0 & \cdots & 0 & X \\ 1 & 0 & & 0 & X \\ 0 & 1 & & 0 & X \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & & 1 & X \end{bmatrix} \qquad [A.12]$$

for $i = 1, \ell$; with dimensions $\mu_i \times \mu_i$. The X's stand for possible nonzero entries. The off-diagonal blocks $\widetilde{A}_{ij}$, $i \neq j$, have the form

$$\widetilde{A}_{ij} = \begin{bmatrix} & & X \\ & & X \\ 0 & & \vdots \\ & & X \\ & & X \end{bmatrix} \qquad [A.13]$$

The transformed output matrix C is

$$\widetilde{C} = \begin{bmatrix} \widetilde{C}_1, \widetilde{C}_2, \ldots, \widetilde{C}_\ell \end{bmatrix} \qquad [A.14]$$

where the $\ell \times \mu_i$ blocks $\widetilde{C}_i$ have the special form

$$\widetilde{C}_i = \begin{bmatrix} 0 & \vdots & \varepsilon_i \end{bmatrix} \qquad [A.15]$$

with

91

$$\varepsilon_j = \begin{cases} 0 & \text{for } j < i \\ 1 & \text{for } j = i \\ X & \text{for } j > i \end{cases} \qquad [A.16]$$

i.e. the entries below the unit entry in the last column of $\widetilde{C}_i$ are not, in general, equal to zero. These nonzero entries can be readily removed, if so desired, by a simple nonsingular transformation of the system outputs.

The matrix $\widetilde{B}$ has no special form.

# APPENDIX B

## NECESSARY PROOFS

### B.1    Proof of Theorem 4.1

Without loss of generality, assume that $\lambda_z < 0.5$. We then need to show that $\lambda_{\hat{z}} < 1$ and hence $\hat{Z} \in H$. Consider the diagonal regular splitting of $Z$ (this is always possible whenever $Z \in H$) as:

$$Z = D + C = \left(I + CD^{-1}\right)D = (I + M)D \in H \qquad [B.1]$$

Note that $I + M \in H$ since multiplication by a nonsingular complex diagonal matrix does not destroy the H-matrix characteristic. Since

$$\rho(M) \leq \rho(M_+) = \rho\left(C_+ D_+^{-1}\right) \equiv \lambda_z < 1 \qquad [B.2]$$

where $\rho(\bullet)$ denotes the spectral radius of $\bullet$, we may write $\hat{Z}$ as the following convergent series:

$$\hat{Z} = D^{-1}(I + M)^{-1} = D^{-1}\left(I + \sum_{k=1}^{\infty}(-M)^k\right) \qquad [B.3]$$

As D is diagonal, $\hat{Z} \in H \Leftrightarrow I + \sum_{k=1}^{\infty}(-M)^k \in H$, or equivalently, if and only if:

93

$$\rho\left(\sum_{k=1}^{\infty}(-M)^k\right)_+ = \lambda_{\hat{z}} < 1.$$ [B.4]

Now,

$$\lambda_{\hat{z}} = \rho\left(\sum_{k=1}^{\infty}(-M)^k\right)_+ \leq \rho\left(\sum_{k=1}^{\infty}M^k\right)_+ \leq \sum_{k=1}^{\infty}\rho(M_+^k) = \sum_{k=1}^{\infty}\lambda_z^k = \frac{\lambda_z}{1-\lambda_z}.$$ [B.5]

Then the infinite series sum [B.5] reduces to:

$$\lambda_{\hat{z}} \leq \frac{\lambda_z}{1-\lambda_z} \Leftrightarrow \frac{\lambda_{\hat{z}}}{1+\lambda_{\hat{z}}} < \lambda_z$$ [B.6]

Thus,

$$\lambda_z < 0.5 \Rightarrow \lambda_{\hat{z}} < 1 \Rightarrow \hat{Z} \in H$$ [B.7]

which completes the proof.

## B.2    Proof of Theorem 4.3

Given that the elements K (precompensator), G (diagonal controller) and F (prefilter) are stable. Then $T_{Y/R}$ is unstable if and only if

$$\det(Z) = \det\left((I+M)(\hat{P}_D + G)\right) = \det(I+M)\det(\hat{P}_D + G)$$ [B.8]

has zeros in the right-hand plane.

The $\det(I + M)$ has no zeros in the right hand plane if $\lambda_p(M) < 1$. In addition,

$$\text{the } \det\left(\hat{P}_D + G\right) = \prod_{i=1}^{n}\left(\left\|\frac{1}{q_{ii}} + g_i\right\|\right) \text{ has no zeros in the right hand plane if each } \left(1 + g_i q_{ii}\right)$$

has no zeros in the right-hand plane. This completes the proof.

# APPENDIX C

## COMPUTER ROUTINES

Contained herein is a set of routines written for Matlab 4.0 that were used to generate the design given in Chapter 5. These routines can be grouped into categories for design, analysis and support. Below is a listing of each routine by category and the page on which it may be found.

```
%==================================================================
%
% Name : Decouple
%
% Purpose:   To generate data for the design of the
%            precompensator
%
% Inputs : None passed to the routine must have system
%          matrices available and an appropriate frequency
%          vector must be set within the routine
%
% Outputs: (qij/qjj) needed for loop shaping
%          bounds for loopshaping
%
% Loopshaping Commands :
%     element 12 : lpshape(w,bnd12,-q12_data(1,:),[])
%     element 21 : lpshape(w,bnd21,-q21_data(1,:),[])
%
% Written by : Zane D. Gastineau
%
%==================================================================

%==================================================================
%  Load system matrices
%==================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn

%==================================================================
%  Frequency range of concern
%==================================================================
w=sort([logspace(-1,2,50),0.5,1.0,1.5,3.0,6.0,8.0]);

%==================================================================
%  Housekeeping
%==================================================================
j=sqrt(-1);
q21_data=[];
q12_data=[];
v=0;

%==================================================================
%  Build up plant and generate loop shaping data.
%==================================================================
for k=15:length(an)-2
k
v=v+1;
%
        [ap,bp,cp,dp]=abcd_n(an,bn,cn,dn,k);
%
        for mi=1:length(w)
                P=cp*(inv(j*w(mi)*eye(size(ap))-ap)*bp)+dp;
                [act,sen]=actblck_w(w(mi));
                L=sen*P*act;
```

97

```
            Lhat=inv(L);
            q12_data(v,mi)=(1/Lhat(1,2))*Lhat(2,2);
            q21_data(v,mi)=(1/Lhat(2,1))*Lhat(1,1);
        end
end

%================================================================
%   Generate Bounds for the precompensator
%
%     |1-kij(qij/qjj)| <= 1
%================================================================
%   Frequencies to Generate bounds at
%
w_bnd=[0.1 0.5 1 3 6 8];
bnd12=genbnds(10,w,w_bnd,1,1,-q12_data,1,0,-q12_data(1,:));
bnd21=genbnds(10,w,w_bnd,1,1,-q21_data,1,0,-q21_data(1,:));
```

```
%==========================================================================
%
%   Name : gdes1.m
%
%   Purpose : Routine to compute loop 1 bounds
%
%   Inputs : None passed to the routine must have system
%            matrices available and an appropriate frequency
%            vector must be set within the routine.
%
%   Outputs : Bounds for use in the loop shaping routine of the
%             QFT toolbox.
%
%   Loopshaping Command :
%             lpshape(w,bnd,qs11(1,:),[])
%
%   Written by : Zane D. Gastineau
%
%==========================================================================
%
%==========================================================================
%    Load plant set
%==========================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn

%==========================================================================
%    Load Frequency set points
%==========================================================================
w=sort([logspace(-1,2,50),0.8,1.0,3.0,6.0,8.0,10,12,20,4.4984]);

%==========================================================================
%    Housekeeping
%==========================================================================
i=sqrt(-1);
v=0;

%==========================================================================
%    Build up Plant over the frequency range
%==========================================================================
for p=15:length(an)-2
v=v+1;
      [ap,bp,cp,dp]=abcd_n(an,bn,cn,dn,p);
      for k=1:length(w)
            P=cp*inv(w(k)*i*eye(size(ap))-ap)*bp+dp;
      [G,KK]=gk_w(w(k));
      [Act,Sen]=actblck_w(w(k));
            L=Sen*P*Act*KK;
            ls11(v,k)=L(1,1);ls12(v,k)=L(1,2);
            ls21(v,k)=L(2,1);ls22(v,k)=L(2,2);
            PKhat=inv(L);
            qs11(v,k) = 1/PKhat(1,1);qs12(v,k) = 1/PKhat(1,2);
            qs21(v,k) = 1/PKhat(2,1);qs22(v,k) = 1/PKhat(2,2);
      end
```

99

```
end

%===========================================================================
%    Set and generate Bounds for QFT design
%===========================================================================
%    Compute Upper (cmu) and Lower (cml) bound over frequency
%===========================================================================
cmu1=freqcp(64,[1 9.6 64],w);
cmu2=freqcp(9,[1 3.6 9],w);
cmu=max(cmu1,cmu2);
cml1=freqcp(169,[1 31.2 169],w);
cml2=freqcp(36,[1 14.4 36],w);
cml=min(cml1,cml2);
Ws=abs([cmu; cml]);
bnd1 = sisobnds(7,w,[0.8 1.0 3.0 6.0],Ws,qs11,[],2);
bnd3 = sisobnds(2,w,[6],1/0.4352,qs11,[],2);
bnd4 = sisobnds(2,w,[8],1/0.51264,qs11,[],2);
bnd=grpbnds(bnd1,bnd3,bnd4);plotbnds(bnd)
```

```
%=======================================================================
%
%   Name : gdes2.m
%
%   Purpose : Routine to compute loop 2 bounds
%
%   Inputs : None passed to the routine must have system
%             matrices available and an appropriate frequency
%             vector must be set within the routine.
%
%   Outputs : Bounds for use in the loop shaping routine of the
%              QFT toolbox.
%
%   Loopshaping Command :
%             lpshape(w,bnd,qs22(1,:),[])
%
%   Written by : Zane D. Gastineau
%
%=======================================================================


%=======================================================================
%   Load plant set
%=======================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn

%=======================================================================
%   Load Frequency set points
%=======================================================================
w=sort([logspace(-1,2,50),0.8,1.0,3.0,6.0,8,10,12,20]);

%=======================================================================
%   Housekeeping
%=======================================================================
i=sqrt(-1);
v=0;


%=======================================================================
%   Build up plant over the frequency range
%=======================================================================
for p=15:length(an)-2
v=v+1
        [ap,bp,cp,dp]=abcd_n(an,bn,cn,dn,p);
        for k=1:length(w)
                P=cp*inv(w(k)*i*eye(size(ap))-ap)*bp+dp;
        [G,KK]=gk_w(w(k));
        [Act,Sen]=actblck_w(w(k));
                L=Sen*P*Act*KK;
                ls11(v,k)=L(1,1);ls12(v,k)=L(1,2);
                ls21(v,k)=L(2,1);ls22(v,k)=L(2,2);
                PKhat=inv(L);
                qs11(v,k) = 1/PKhat(1,1);qs12(v,k) = 1/PKhat(1,2);
                qs21(v,k) = 1/PKhat(2,1);qs22(v,k) = 1/PKhat(2,2);
        end
```

```
end

%=================================================================
%    Set and generate Bounds for QFT design
%=================================================================
%    Compute Upper (cmu) and Lower (cml) bound over frequency
%=================================================================
cmu=freqcp(36,[1 60 36],w);
cml=freqcp(16,[1 40 16],w);
Ws=abs([cmu; cml]);
bnd1 = sisobnds(7,w,[0.8 1.0 3.0 6.0],Ws,qs22,[],2);
bnd3 = sisobnds(2,w,[6],1/0.4352,qs22,[],2);
bnd4 = sisobnds(2,w,[8],1/0.5126,qs22,[],2);
bnd=grpbnds(bnd1,bnd3,bnd4);
plotbnds(bnd)
```

```
%=====================================================================
%   Name : pfdes1.m
%
%   Purpose : To generate required data for the design of the
%             prefilter
%
%   Inputs : None passed to the routine must have system matrices
%            available and an appropriate frequency vector must
%            be set within the routine.
%
%   Outputs : L(i,j) for prefilter design loopshaping.
%
%   Loopshaping Command :
%               pfshape(7,w,w,Ws,L11(1,:),[],G11(1,:))
%
%   Written by : Zane D. Gastineau
%=====================================================================


%=====================================================================
%    Load plant set
%=====================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn


%=====================================================================
%    Load Frequency set points
%=====================================================================
w=sort([logspace(-1,2,50),0.8,1.0,3.0,6.0,10,12,20,4.4984]);


%=====================================================================
%   Housekeeping
%=====================================================================
i=sqrt(-1);
v=0;


%=====================================================================
%    Build up Plant over the frequency range
%=====================================================================
for p=15:length(an)-2
v=v+1
        [ap,bp,cp,dp]=abcd_n(an,bn,cn,dn,p);
        for k=1:length(w)
                P=cp*inv(w(k)*i*eye(size(ap))-ap)*bp+dp;
        [G,KK]=gk_w(w(k));
        G11(v,k)=G(1,1);
                [Act,Sen]=actblck_w(w(k));
                L=Sen*P*Act*KK;
                L11(v,k)=L(1,1);L12(v,k)=L(1,2);
                L21(v,k)=L(2,1);L22(v,k)=L(2,2);
        end
end
```

103

```
%=============================================================
%    Compute Upper (cmu) and Lower (cml) bound over frequency
%=============================================================
cmu1=freqcp(64,[1 9.6 64],w);
cmu2=freqcp(9,[1 3.6 9],w);
cmu=max(cmu1,cmu2);
cml1=freqcp(169,[1 31.2 169],w);
cml2=freqcp(36,[1 14.4 36],w);
cml=min(cml1,cml2);
Ws=abs([cmu; cml]);
```

```
%====================================================================
%  Name : pfdes2.m
%
%  Purpose : To generate required data for the design of the
%            prefilter
%
%  Inputs : None passed to the routine must have system matrices
%           available and an appropriate frequency vector must
%           be set within the routine.
%
%  Outputs : L(i,j) for prefilter design loopshaping.
%
%  Loopshaping Command :
%              pfshape(7,w,w,Ws,L22(1,:),[],G22(1,:))
%
%  Written by : Zane D. Gastineau
%
%====================================================================


%====================================================================
%   Load plant set
%====================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn

%====================================================================
%   Load Frequency set points
%====================================================================
w=sort([logspace(-1,2,50),0.8,1.0,3.0,6.0,10,12,20,4.4984]);

%====================================================================
%  Housekeeping
%====================================================================
i=sqrt(-1);
v=0;

%====================================================================
%   Build up Plant over the frequency range
%====================================================================
for p=15:length(an)-2
v=v+1
      [ap,bp,cp,dp]=abcd_n(an,bn,cn,dn,p);
      for k=1:length(w)
            P=cp*inv(w(k)*i*eye(size(ap))-ap)*bp+dp;
      [G,KK]=gk_w(w(k));
      G22(v,k)=G(2,2);
      [Act,Sen]=actblck_w(w(k));
            L=Sen*P*Act*KK;
            L11(v,k)=L(1,1);L12(v,k)=L(1,2);
            L21(v,k)=L(2,1);L22(v,k)=L(2,2);
      end
end
```

105

```
%=====================================================================
%    Compute Upper (cmu) and Lower (cml) bound over frequency
%=====================================================================
cmu=freqcp(16,[1 16 16],w);
cml=freqcp(36,[1 24 36],w);
Ws=abs([cmu; cml]);
```

```
%================================================================
%
%  Name : Interact.m
%
%  Purpose : Routine to compute the interaction index
%            for the system
%
%  Inputs : None passed to the routine must have system
%           matrices available and an appropriate frequency
%           vector must be set within the routine.
%
%  Outputs : Plot of the interaction index for either the
%            direct or for the inverse system.
%
%  Written by : Zane D. Gastineau
%
%================================================================

%================================================================
%  Load plant set
%================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn

%================================================================
%  Housekeeping
%================================================================
v=0;
i=sqrt(-1);

%================================================================
%  Load Frequency Data
%================================================================
w=sort([logspace(-2,3,50),0.5,1.0,3.0,6.0,8.0]);

%================================================================
%  Compute interaction index
%================================================================
for k=15:size(an,1)-2
k
v=v+1;
        [ac,bc,cc,dc]=abcd_n(an,bn,cn,dn,k);
                for q=1:length(w)
                        [act,sen]=actblck_w(w(q));
                        %ke=eye(2);   % Use this for plant interaction index
                [G,ke]=gk_w(w(q));
                P=cc*inv(w(q)*i*eye(3)-ac)*bc+dc;
                v1=sen*P*act*ke;
                %v1=inv(P); %Use this for the inverse plant.
                v1d=diag(diag(v1));
                        vlo=v1-v1d;
                        M1=vlo*inv(v1d);
                        [lambda,v0]=perron(M1);
                z1(v,q)=lambda;
```

107

```
              end
end

%================================================================
%   Plot Interaction Index
%================================================================
for q=1:v
              semilogx(w,z1(q,:))
              hold on
       end
grid
xlabel('frequency (rad/s)')
ylabel('Interaction Index')
```

```
%=====================================================================
%
%   Name : mybodes.m
%
%   Pupose : To plot magnitude vs frequency for different
%            functions.
%
%   Inputs : None passed to the routine must have system
%            matrices available and an appropriate frequency
%            vector set within the routine.  Actuator,
%            Sensor, Controller and Precompensator must be
%            defined outside the routine to be called
%
%   Outputs : Magnitude responses for the Plant, Plant+
%             precompensator, Plant+precompensator+controller,
%             closed loop system and closed loop system with
%             prefilter.
%
%   Written by : Zane D. Gastineau
%
%=====================================================================

%=====================================================================
%   Select which plot you want.
%=====================================================================
disp('Select the desired plot:')
disp(' ')
disp('  1.   PIant (P)')
disp('  2.   Plant+Precompensator (PK)')
disp('  3.   Plant+Precompensator+Controller (L)')
disp('  4.   Closed loop system [inv(I+L)*L]')
disp('  5.   Closed loop system with prefilter [inv(I+L)*LF]')
disp(' ')
in = input('              Enter choice : ');

%=====================================================================
%    Load plant set
%=====================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn

%=====================================================================
%    Load Frequency set points
%=====================================================================
w=sort([logspace(-1,2,50),0.8,1.0,3.0,6.0,10,12,20,4.4984]);

%=====================================================================
%    Build up Plant over the frequency range
%=====================================================================
for p=15:length(an)-2
p
        [ap,bp,cp,dp]=abcd_n(an,bn,cn,dn,p);
        for k=1:length(w)
        Plant=cp*inv(w(k)*i*eye(size(ap))-ap)*bp+dp;
```

```
            [Act,Sen]=actblck_w(w(k));
            P=Sen*Plant*Act;
            P11(p,k)=P(1,1);P12(p,k)=P(1,2);
            P21(p,k)=P(2,1);P22(p,k)=P(2,2);
            [G,KK]=gk_w(w(k));
            PK=P*KK;
            PK11(p,k)=PK(1,1);PK12(p,k)=PK(1,2);
                PK21(p,k)=PK(2,1);PK22(p,k)=PK(2,2);
                PKhat=inv(PK);
            qs11(p,k)=1/PKhat(1,1);qs12(p,k)=1/PKhat(1,2);
            qs21(p,k)=1/PKhat(2,1);qs22(p,k)=1/PKhat(2,2);
            L=PK*G;
            L11(p,k)=L(1,1);L12(p,k)=L(1,2);
            L21(p,k)=L(2,1);L22(p,k)=L(2,2);
            T=inv(eye(2)+L)*L;
            T11(p,k)=T(1,1);T12(p,k)=T(1,2);
            T21(p,k)=T(2,1);T22(p,k)=T(2,2);
            [F]=pfilter_w(w(k));
            TF=T*F;
            TF11(p,k)=TF(1,1);TF12(p,k)=TF(1,2);
            TF21(p,k)=TF(2,1);TF22(p,k)=TF(2,2);
        end
end
%====================================================================
%    Generate Magnitude vs frequency plots
%====================================================================
%    compute individual magnitude responses
%====================================================================
for k=15:29
    if in == 1
        subplot(2,2,1);semilogx(w,db(P11(k,:)));grid;hold on;
        subplot(2,2,2);semilogx(w,db(P12(k,:)));grid;hold on;
        subplot(2,2,3);semilogx(w,db(P21(k,:)));grid;hold on;
        subplot(2,2,4);semilogx(w,db(P22(k,:)));grid;hold on;
    elseif in == 2
        subplot(2,2,1);semilogx(w,db(PK11(k,:)));grid;hold on;
        subplot(2,2,2);semilogx(w,db(PK12(k,:)));grid;hold on;
        subplot(2,2,3);semilogx(w,db(PK21(k,:)));grid;hold on;
        subplot(2,2,4);semilogx(w,db(PK22(k,:)));grid;hold on;
    elseif in == 3
        subplot(2,2,1);semilogx(w,db(L11(k,:)));grid;hold on;
        subplot(2,2,2);semilogx(w,db(L12(k,:)));grid;hold on;
        subplot(2,2,3);semilogx(w,db(L21(k,:)));grid;hold on;
        subplot(2,2,4);semilogx(w,db(L22(k,:)));grid;hold on;
    elseif in == 4
        subplot(2,2,1);semilogx(w,db(T11(k,:)));grid;hold on;
        subplot(2,2,2);semilogx(w,db(T12(k,:)));grid;hold on;
        subplot(2,2,3);semilogx(w,db(T21(k,:)));grid;hold on;
        subplot(2,2,4);semilogx(w,db(T22(k,:)));grid;hold on;
    elseif in == 5
        subplot(2,2,1);semilogx(w,db(TF11(k,:)));grid;hold on;
        subplot(2,2,2);semilogx(w,db(TF12(k,:)));grid;hold on;
        subplot(2,2,3);semilogx(w,db(TF21(k,:)));grid;hold on;
        subplot(2,2,4);semilogx(w,db(TF22(k,:)));grid;hold on;
    end
end
```

110

```
%==========================================================================
%
%   Name : mysteps.m
%
%   Purpose : To generate time response plots of the plant,
%             the closed loop system and the closed loop
%             system with the prefilter.
%
%   Inputs : None passed to the routine.  Must have the
%            actuator and sensor routines predefined.
%
%   Outputs : Plot of plant and closed loop system step
%             responses
%
%   Written by : Zane D. Gastineau
%
%==========================================================================


%==========================================================================
%   Load system matrices
%==========================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn


%==========================================================================
%   Select proper plot routine
%==========================================================================
disp('Select the desired Plot:')
disp(' ')
disp(' 1. Plant (P)')
disp(' 2. Closed Loop System [inv(I+L)*L]')
disp(' 3. Closed Loop System with Prefilter [inv(I+L)*L*F]')
disp(' ')
in = input('                Enter Choice : ');


%==========================================================================
%   Housekeeping
%==========================================================================
hold off
clf


%==========================================================================
%   Load appropriate State Space Matrices
%==========================================================================
[act,sen]=actblck;
[G,KK]=gk;
[F]=pfilter;
[Aact,Bact,Cact,Dact]=branch(act);
[Asen,Bsen,Csen,Dsen]=branch(sen);
[Ag,Bg,Cg,Dg]=branch(G);
[Ak,Bk,Ck,Dk]=branch(KK);
[Fa,Fb,Fc,Fd]=branch(F);
```

111

```
%=================================================================
%  Build up System Matrices
%=================================================================
for p=16:size(an,1)-2
p
[A,B,C,D]=abcd_n(an,bn,cn,dn,p);
[A,B,C,D]=series(Aact,Bact,Cact,Dact,A,B,C,D);
[A,B,C,D]=series(A,B,C,D,Asen,Bsen,Csen,Dsen);
  if in == 1
     for m=1:2
      t=[0:0.1:5];
         [y]=step(A,B,C,D,m,t);
            for n=1:2
              str=['22' num2str(m+(n-1)*2)];
          subplot(str);grid;
              plot(t,y(:,n))
              hold on
       end
      end
            subplot(221);title('Delta Fuel Flow');ylabel('Delta
Thrust');
            subplot(222);title('Delta Nozzle Area');
            subplot(223);ylabel('Delta Stall Margin');xlabel('time
(sec)');
      subplot(224);xlabel('time (sec)');
    elseif in == 2
      [A,B,C,D]=series(Ak,Bk,Ck,Dk,A,B,C,D);
            [A,B,C,D]=series(Ag,Bg,Cg,Dg,A,B,C,D);
      [A,B,C,D]=cloop(A,B,C,D,-1);
      for m=1:2
         t=[0:0.1:10];
         [y]=step(A,B,C,D,m,t);
                for n=1:2
                  str=['22' num2str(m+(n-1)*2)];
                  subplot(str);grid;
                  plot(t,y(:,n))
                  hold on
          end
         end
      subplot(221);title('Delta Thrust Demand');
                  ylabel('Delta Thrust');
            subplot(222);title('Delta Stall Margin Demand');
      subplot(223);ylabel('Delta Stall Margin');
                  xlabel('time (sec)');
      subplot(224);xlabel('time (sec)');
       elseif in == 3
      [A,B,C,D]=series(Ak,Bk,Ck,Dk,A,B,C,D);
            [A,B,C,D]=series(Ag,Bg,Cg,Dg,A,B,C,D);
      [A,B,C,D]=cloop(A,B,C,D,-1);
      [A,B,C,D]=series(Fa,Fb,Fc,Fd,A,B,C,D);
      for m=1:2
         t=[0:0.1:10];
                  [y]=step(A,B,C,D,m,t);
                        for n=1:2
                           str=['22' num2str(m+(n-1)*2)];
                           subplot(str);grid;
```

112

```
                        plot(t,y(:,n))
                        hold on
                end
            end
      subplot(221);title('Delta Thrust Demand');
                  ylabel('Delta Thrust');
          subplot(222);title('Delta Stall Margin Demand');
      subplot(223);ylabel('Delta Stall Margin');
                  xlabel('time (sec)');
      subplot(224);xlabel('time (sec)');
  end
end
```

```
%==================================================================
%
%   Name : Pathconn.m
%
%   Purpose : Routine for investigating the path connectedness
%             of one plant to another.
%
%   Inputs : None supplied to the routine.  The user will be
%            prompted for the nominal plant #.  Then the user
%            can test any other plant within the plant set
%            one at a time.
%
%   Outputs : Plot of ratio of determinants.  Indicates path
%             connectedness.
%
%   Written by : Zane D. Gastineau
%                               6909 Custer Road #2104
%                    Plano, Tx 75023
%                    (972)491-3776
%==================================================================


%==================================================================
%   load state-space models
%==================================================================
load c:\matlab\turbojet\normmat\sls_an
load c:\matlab\turbojet\normmat\sls_bn
load c:\matlab\turbojet\normmat\sls_cn
load c:\matlab\turbojet\normmat\sls_dn


%==================================================================
%   Load the nominal plant
%==================================================================
in1 = input('Enter the number for the nominal plant : ');
[anom,bnom,cnom,dnom]=abcd_n(an,bn,cn,dn,in1);


%==================================================================
%   Load desired frequency range
%==================================================================
w=logspace(-1,2,50);


%==================================================================
%   Select display method
%==================================================================
disp(' ')
disp('Select Method of Display')
disp(' ')
disp('  1. User defined plant')
disp('  2. Loop through all plants')
disp(' ')
in2 = input('                    Enter Method : ');
if in2 == 1
   disp(' ')
   in3 = input('Enter Desired Plant to Check : ');
   z=[];
      [ac,bc,cc,dc]=abcd_n(an,bn,cn,dn,k);
              for q=1:length(w)
```

114

```
                    v1=cnom*inv(w(q)*i*eye(3)-anom)*bnom+dnom;
                    v2=cc*inv(w(q)*i*eye(3)-ac)*bc+dc;
                    r1=det(v1);
                    r2=det(v2);
                    z=[z;r2/r1];
        end
        for q=1:length(w)
                    plot(real(z),imag(z))
                    str=['plant ' num2str(k)];
                    title(str)
                    ylabel('imag(z)')
                    xlabel('real(z)')
            end
elseif in2 == 2
        for k=1:size(an,1) %use to loop through plants
        k
        z=[];
        [ac,bc,cc,dc]=abcd_n(an,bn,cn,dn,k);
            for q=1:length(w)
                    v1=cnom*inv(w(q)*i*eye(3)-anom)*bnom+dnom;
                    v2=cc*inv(w(q)*i*eye(3)-ac)*bc+dc;
                    r1=det(v1);
                    r2=det(v2);
                    z=[z;r2/r1];
        end
        for q=1:length(w)
                    plot(real(z),imag(z))
                    str=['plant ' num2str(k)];
                    title(str)
                    ylabel('imag(z)')
                    xlabel('real(z)')
            end
            pause
    end
end
```

115

```matlab
function [A,B,C,D]=abcd_n(a,b,c,d,k)
%===================================================================
%
%   Name : abcd.m
%
%   Purpose : Function returns kth plant of full set of plants.
%             All outputs available.
%
%   Inputs :  Must supply system matrices as large matrices where
%             each plant makes up one row.  User must also supply
%             which plant they are interested in.
%
%   Outputs : The desired plant in State Space Form
%
%   Written by : Zane D. Gastineau
%
%===================================================================

A = [a(k,1)  a(k,2)  a(k,3)
     a(k,4)  a(k,5)  a(k,6)
     a(k,7)  a(k,8)  a(k,9)];

B = [ b(k,1)  b(k,2)   b(k,3)   b(k,4)   b(k,5)
      b(k,6)  b(k,7)   b(k,8)   b(k,9)  b(k,10)
     b(k,11) b(k,12)  b(k,13)  b(k,14)  b(k,15)];

C = [ c(k,1)   c(k,2)   c(k,3)
      c(k,4)   c(k,5)   c(k,6)
      c(k,7)   c(k,8)   c(k,9)
     c(k,10)  c(k,11)  c(k,12)
     c(k,13)  c(k,14)  c(k,15)
     c(k,16)  c(k,17)  c(k,18)
     c(k,19)  c(k,20)  c(k,21)
     c(k,22)  c(k,23)  c(k,24)
     c(k,25)  c(k,26)  c(k,27)];

D = [ d(k,1)   d(k,2)   d(k,3)   d(k,4)   d(k,5)
      d(k,6)   d(k,7)   d(k,8)   d(k,9)  d(k,10)
     d(k,11)  d(k,12)  d(k,13)  d(k,14)  d(k,15)
     d(k,16)  d(k,17)  d(k,18)  d(k,19)  d(k,20)
     d(k,21)  d(k,22)  d(k,23)  d(k,24)  d(k,25)
     d(k,26)  d(k,27)  d(k,28)  d(k,29)  d(k,30)
     d(k,31)  d(k,32)  d(k,33)  d(k,34)  d(k,35)
     d(k,36)  d(k,37)  d(k,38)  d(k,39)  d(k,40)
     d(k,41)  d(k,42)  d(k,43)  d(k,44)  d(k,45)];
```

116

```
function [A,B,C,D]=abcd_n(a,b,c,d,k)
%=====================================================================
%      .
%  Name : abcd_n.m
%
%  Purpose : Function returns kth plant of full set of plants.
%              Thrust and Stall Margin Outputs available only the
%              Fuel Flow and Nozzle Area are available for input.
%
%  Inputs :  Must supply system matrices as large matrices where
%              each plant makes up one row.  User must also supply
%              which plant they are interested in.
%
%  Outputs : The desired plant in State Space Form
%
%  Written by : Zane D. Gastineau
%
%=====================================================================

A = [a(k,1)  a(k,2)  a(k,3)
     a(k,4)  a(k,5)  a(k,6)
     a(k,7)  a(k,8)  a(k,9)];

B = [ b(k,1)   b(k,5)
      b(k,6)  b(k,10)
     b(k,11)  b(k,15)];

C = [c(k,1)  c(k,2)  c(k,3)
     c(k,4)  c(k,5)  c(k,6)];

D = [d(k,1)   d(k,5)
     d(k,6)  d(k,10)];
```

117

```
function [act,sen]=actblck
%================================================================
%

%   Name : Actblck.m
%
%   Purpose : To create state space model of the actuator block
%             and the sensor block for time domain analysis.  All
%             computational delays are included.
%
%   Inputs : None supplied.
%
%   Outputs : State Space models of the actuators and sensors.
%
%   Written by : Zane D. Gastineau
%                6906 Custer Rd #2104
%                Plano, TX 75023
%                (937)491-3776
%
%================================================================


%================================================================
%   Actuator Model #1
%   Fuel Flow Model
%================================================================
num1=[18];
den1=[1 18];
%
%   Total 60msec delay in this channel
%   40 msec for bypass dynamics and combustion delay
%   20 msec for computational delay
%
[num2,den2]=pade(0.02,4);% 4th order approx. to 20msec delay
[num,den]=pade(0.04,4);% 4th order approx. to 40msec delay
[a11,b11,c11,d11]=tf2ss(num1,den1);
[a22,b22,c22,d22]=tf2ss(num,den);
[a33,b33,c33,d33]=tf2ss(num2,den2);
%
%   Create state space actuator model
%
[a1,b1,c1,d1]=series(a33,b33,c33,d33,a11,b11,c11,d11);
[a1,b1,c1,d1]=series(a1,b1,c1,d1,a22,b22,c22,d22);


%================================================================
%   Actuator Model #2
%   A8 Actuator
%================================================================
num1=[18];
den1=[1 18];
%
%   Computational time delay
%   20 milliseconds
%
[num,den]=pade(0.02,4); %4th order approx. to 20msec delay
[a11,b11,c11,d11]=tf2ss(num1,den1);
[a22,b22,c22,d22]=tf2ss(num,den);
```

118

```
%
%  Create state space model of actuator
%
[a2,b2,c2,d2]=series(a22,b22,c22,d22,a11,b11,c11,d11);


%===================================================================
%  Sensor Model
%  Used for both Thrust and Stall Margin
%  Feedback.  A time delay is used to represent
%  the computational time delay of the computer
%  to compute Thrust and Stall Margin.  This
%  value is set to 20 milliseconds.
%===================================================================
num=[25];
den=[1 25];
[num2,den2]=pade(0.02,4); %4th order approx. to 20 msec delay
[a1,b1,c1,d1]=tf2ss(num,den);
[as,bs,cs,ds]=tf2ss(num2,den2);
[as,bs,cs,ds]=series(a1,b1,c1,d1,as,bs,cs,ds);
%
%  Now make state space model for the actuator/delay block:
%
[Aact,Bact,Cact,Dact]=append(a1,b1,c1,d1,a2,b2,c2,d2);
%
%  Now make state space model for the sensors
%
[Asen,Bsen,Csen,Dsen]=append(as,bs,cs,ds,as,bs,cs,ds);


%===================================================================
%  Return Matrices in Compact Form. Routines in Robust Control
%  Toolbox.
%===================================================================
act=mksys(Aact,Bact,Cact,Dact);
sen=mksys(Asen,Bsen,Csen,Dsen);
```

119

```
function [act,sen]=actblck_w(w)
%==================================================================
%
%   Name : Actblck_w.m
%
%   Purpose : To return complex evaluation of acturators and
%             sensors for frequency domain analysis.  All
%             computational delays are included.
%
%   Inputs : User supplied frequency value in radians/second.
%            The frequency value must be a scalar.
%
%   Outputs : Complex number evaluations for the actuators and
%             sensors in matrix form.
%
%   Written by : Zane D. Gastineau
%                6906 Custer Rd #2104
%                Plano, TX 75023
%                (937)491-3776
%
%==================================================================


%==================================================================
%   Houskeeping
%==================================================================
s=sqrt(-1)*abs(w);   % if w is complex
[nr,nc]=size(w);
if nr > 1|nc > 1
   disp('frequency value must be a scalar')
else
%==================================================================
%   Actuator Model #1
%   Fuel Flow Model
%
%   Total 60msec delay in this channel
%   40 msec for bypass dynamics and combustion delay
%   20 msec for computational delay
%==================================================================
act1=(18./(s+18)).*exp(-s*60e-3);


%==================================================================
%   Actuator Model #2
%   A8 Actuator
%
%   Computational time delay
%   20 milliseconds
%==================================================================
act2=(18./(s+18)).*exp(-s*20e-3);
```

120

```
%===================================================================
%   Sensor Model
%   Used for both Thrust and Stall Margin
%   Feedback.  A time delay is used to represent
%   the computational time delay of the computer
%   to compute Thrust and Stall Margin.  This
%   value is set to 20 milliseconds.
%===================================================================
sen1=exp(-s*20e-3);

act=diag([act1 act2]);
sen=diag([sen1 sen1]);
end
```

```
function [cont,decple]=gk
%=================================================================
%
%   Name : gk.m
%
%   Purpose : Routine to return state space form of controller
%             and decoupler
%
%   Inputs : No inputs required
%
%   Outputs : State space models of the precompensator and
%             controller.
%
%   Written by : Zane D. Gastineau
%
%=================================================================


%=================================================================
%   Diagonal Controller
%=================================================================
g11num=7*conv([1/100 0.15 1],conv([1/4.201 1],[1/4.958 1]));
g11dena=conv([1 0],conv([1/5 1],conv([1/8.560 1],[1/69.06 1])));
g11den=conv(g11dena,[1/120 1]);
g22num=0.8*conv([1/0.7025 1],[1/2.38 1]);
g22den=conv([1 0],conv([1/0.5124 1],[1/6.723 1]));
[g11a,g11b,g11c,g11d]=tf2ss(g11num,g11den);
[g22a,g22b,g22c,g22d]=tf2ss(g22num,g22den);
[Ag,Bg,Cg,Dg]=append(g11a,g11b,g11c,g11d,g22a,g22b,g22c,g22d);


%=================================================================
%   Precompensator
%=================================================================
k11num=1;
k11den=1;
k22num=1;
k22den=1;
k12num=0.05*conv([1/0.1034 1],[1/2.144 1]);
k12den=conv([1/1.547 1],[1/29.86 1]);
k21num=0.005*[1/0.02 1];
k21den=[1/2.087 1];
[k11a,k11b,k11c,k11d]=tf2ss(k11num,k11den);
[k22a,k22b,k22c,k22d]=tf2ss(k22num,k22den);
[k12a,k12b,k12c,k12d]=tf2ss(k12num,k12den);
[k21a,k21b,k21c,k21d]=tf2ss(k21num,k21den);
sk1=size(k11a);sk2=size(k12a);sk3=size(k21a);sk4=size(k22a);
Ako=[k12a zeros(sk2(1),sk3(2));
     zeros(sk3(1),sk2(2)) k21a];
Bko=[zeros(sk2(1),1) k12b
     k21b  zeros(sk3(1),1)];
Cko=[k12c zeros(1,sk3(2))
     zeros(1,sk2(2)) k21c];
Dko=[0 k12d;k21d 0];
[Akd,Bkd,Ckd,Dkd]=append(k11a,k11b,k11c,k11d,k22a,k22b,k22c,k22d);
[Ak,Bk,Ck,Dk]=parallel(Akd,Bkd,Ckd,Dkd,Ako,Bko,Cko,Dko);
```

```
%=====================================================================
%   Return in compact form
%   Routines in Robust control toolbox
%=====================================================================
cont=mksys(Ag,Bg,Cg,Dg);
decple=mksys(Ak,Bk,Ck,Dk);
```

```matlab
function [ctrl,dec]=gk_w(w)
%=========================================================================
%
%   Name : gk_w.m
%
%   Purpose : Routine to return frequency response of controller
%             and decoupler
%
%   Inputs : supply desired frequency to compute at
%
%   Outputs : Complex matrix representing the precompensator and
%             Controller.
%
%   Written by : Zane D. Gastineau
%
%=========================================================================

%=========================================================================
%   Housekeeping
%=========================================================================

s=sqrt(-1).*w;

%=========================================================================
%   Controller responses
%=========================================================================

g11numa=7.*((s/100).^2+(2*0.75/10).*s+1);
g11numb=(s/4.201+1).*(s/4.958+1);
g11dena=s.*(s/5+1).*(s/8.560+1);
g11denb=(s/69.06+1).*(s/120+1);

g11=(g11numa.*g11numb)./(g11dena.*g11denb);

g22num=0.8.*(s/0.7025+1).*(s/2.38+1);
g22den=s.*(s/0.5124+1).*(s/6.723+1);

g22=g22num./g22den;

%=========================================================================
%   Precompensator Responses
%=========================================================================

k11=1;k22=1;

k12num=0.05.*(s/0.1034+1).*(s/2.144+1);
k12den=(s/1.547+1).*(s/29.86+1);

k12=k12num./k12den;

k21num=0.005.*(s/0.02+1);
k21den=(s/2.087+1);

k21=k21num./k21den;
```

```
%=====================================================================
%   returned matrices
%=====================================================================

ctrl=[g11 0;0 g22];
dec=[k11 k12;k21 k22];
```

```
function [F]=pfilter
%===============================================================
%
%   Name : pfilter.m
%
%   Purpose : Routine to return state space model of the
%             prefilter
%
%   Inputs : No inputs required.
%
%   Outputs : State space models of the prefilter.
%
%   Written by : Zane D. Gastineau
%===============================================================
f11num=1;
f11den=[1/3.963 1];
%
f22num=conv([1/2.703 1],[1/11.96 1]);
f22den=conv([1/1.32 1],[1/3.139 1]);
%
[f11a,f11b,f11c,f11d]=tf2ss(f11num,f11den);
[f22a,f22b,f22c,f22d]=tf2ss(f22num,f22den);
%
[Fa,Fb,Fc,Fd]=append(f11a,f11b,f11c,f11d,f22a,f22b,f22c,f22d);

%===============================================================
%   Return in Compact form
%   Routines in Robust control toolbox
%===============================================================
F=mksys(Fa,Fb,Fc,Fd);
```

```matlab
function [F]=pfilter_w(w)
%================================================================
%
%  Name : pfilter_w
%
%  Purpose : Routine to return frequency response of the
%            prefilter.
%
%  Inputs : Supply desired frequency to compute at.
%
%  Outputs : Complex matrix representing the prefilter.
%
%  Written by : Zane D. Gastineau
%================================================================

%================================================================
%  Housekeeping
%================================================================
s=sqrt(-1).*w;

%================================================================
%  Prefilter responses
%================================================================
f11num=1;
f11den=(s/3.963+1);

f11=f11num./f11den;

f22num=(s/2.703+1).*(s/11.96+1);
f22den=(s/1.32+1).*(s/3.139+1);

f22=f22num./f22den;

%================================================================
%  Returned Matrices
%================================================================
F=[f11 0;0 f22];
```

```matlab
function [lambda, v]=perron(Z)
%================================================================
%
%   Name : Perron.m
%
%   Purpose : Extracts the Perron root if it exists
%
%   Inputs : Complex Matrix
%
%   Outputs : The Perron root
%
%================================================================
Z=abs(Z);

if (nargout==1)
      lambda=real(max(eig(Z)));
else
      [X,L]=eig(Z);
      L=diag(L);
      [L,index]=sort(L);
      lambda=real(L(length(L)));
      x=real(X(:,index(length(L))));
      [Y,L]=eig(Z');
      L=diag(L);
      [L,index]=sort(L);
      y=real(Y(:,index(length(L))));

      v=(x*y')/(y'*x);
end
```

# APPENDIX D

## NORMALIZED PLANT MODELS

The following pages contain the normalized linear model data for the turbojet engine used for this work. This data was produced from a nonlinear simulation built in the Math Works Simulink simulation environment using the linearization tools that were available. All the points represent the engine at sea level static conditions and the points are generated at values of corrected speed from idle (70% corrected speed) to full speed (100% corrected speed). These models do not include the actuators and sensors. The states, inputs and outputs are given below.

$$x = \begin{bmatrix} \text{Rotational Speed (N)} \\ \text{Combustor Pressure } (P_4) \\ \text{Tailpipe Pressure } (P_6) \end{bmatrix}$$

$$u = \begin{bmatrix} \text{Fuel Flow } (w_f) \\ \text{Nozzle Area } (A_8) \end{bmatrix}$$

$$y = \begin{bmatrix} \text{Thrust (Fn)} \\ \text{Stall Margin (SM)} \\ \text{Rotational Speed (N)} \\ \text{Compressor Exit Temp. } (T_3) \\ \text{Turbine Exit Temp. } (T_5) \\ \text{Turbine Exit Press. } (P_5) \\ \text{Compressor Exit Press. } (P_3) \\ \text{Turbine Inlet Temp. } (T_4) \\ \text{\% Corrected Speed } (\% N_{corr}) \end{bmatrix}$$

129

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 70 | -3.4508 | 6.3340 | -10.4888 | 2.1432 | 0.0000 | 0.0098 | -0.0051 | 1.6550 | 0.0119 | 0.0426 |
| | | 137.3748 | -327.3311 | 205.6166 | 167.6608 | 0.0000 | 2.1684 | -2.8139 | 0.0000 | 0.0000 | 0.0000 |
| | | 6.2851 | 112.0656 | -1632.6121 | 7.2241 | -96.5396 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.2360 | 0.3721 | 1.8660 | 0.0000 | 0.0000 |
| | | | | | | | -0.0489 | -0.9309 | 1.0200 | 4.0718 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.4405 | 0.5379 | 0.0000 | 4.4408 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 71 | -3.6338 | 6.3794 | -10.6017 | 2.1100 | 0.0000 | 0.0112 | -0.0067 | 1.4991 | 0.0134 | 0.0563 |
| | | 138.4612 | -341.9738 | 207.2184 | 167.7833 | 0.0000 | 2.0370 | -2.9081 | 0.0000 | 0.0000 | 0.0000 |
| | | 6.3846 | 112.2967 | -1318.5529 | 7.4987 | -112.2576 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.2708 | 0.4195 | 1.8643 | 0.0000 | 0.0000 |
| | | | | | | | -0.0283 | -0.8511 | 1.0200 | 3.9626 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.4132 | 0.6561 | 0.0000 | 4.3210 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 3 | 72 | -3.0032 | 6.5395 | -10.7089 | 2.0845 | 0.0000 | 0.0109 | -0.0083 | 1.4282 | 0.0144 | 0.0669 |
| | | 126.3315 | -357.1684 | 208.4973 | 167.8944 | 0.0000 | 1.8114 | -3.0039 | 0.0000 | 0.0000 | 0.0000 |
| | | 5.8927 | 112.2675 | -1168.6773 | 7.7429 | -123.6855 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1665 | 0.4431 | 1.8680 | 0.0000 | 0.0000 |
| | | | | | | | -0.1135 | -0.7956 | 1.0200 | 3.8697 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.4802 | 0.7523 | 0.0000 | 4.2205 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

No. %N | A | B | C | D

**No. 4, %N 73**

A:
| | | |
|---|---|---|
| -3.0797 | 6.7650 | -10.7836 |
| 131.5597 | -368.6180 | 209.6736 |
| 6.1928 | 112.3230 | -1046.6744 |

B:
| | |
|---|---|
| 2.0685 | 0.0000 |
| 167.9878 | 0.0000 |
| 7.8898 | -134.9021 |

| | C | | D | |
|---|---|---|---|---|
| 0.0121 | -0.0097 | 1.3644 | 0.0154 | 0.0781 |
| 1.8082 | -3.0284 | 0.0000 | 0.0000 | 0.0000 |
| 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.1594 | 0.4314 | 0.0000 | 0.0000 | 0.0000 |
| -0.1371 | -0.7811 | 1.8690 | 3.7848 | 0.0000 |
| 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| -0.5199 | 0.7877 | 0.0000 | 4.1302 | 0.0000 |
| 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**No. 5, %N 74**

A:
| | | |
|---|---|---|
| -3.1671 | 6.9902 | -10.8526 |
| 137.4186 | -380.1691 | 210.8110 |
| 6.5292 | 112.3357 | -951.7744 |

B:
| | |
|---|---|
| 2.0546 | 0.0000 |
| 168.0781 | 0.0000 |
| 8.0192 | -145.8120 |

| | C | | D | |
|---|---|---|---|---|
| 0.0134 | -0.0112 | 1.3070 | 0.0163 | 0.0898 |
| 1.8126 | -3.0503 | 0.0000 | 0.0000 | 0.0000 |
| 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.1521 | 0.4199 | 0.0000 | 0.0000 | 0.0000 |
| -0.1612 | -0.7653 | 1.8688 | 3.7020 | 0.0000 |
| 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| -0.5620 | 0.8241 | 0.0000 | 4.0426 | 0.0000 |
| 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

**No. 6, %N 75**

A:
| | | |
|---|---|---|
| -3.2726 | 6.9617 | -10.9162 |
| 144.2717 | -378.1445 | 211.9083 |
| 6.9203 | 112.9730 | -876.0424 |

B:
| | |
|---|---|
| 2.0426 | 0.0000 |
| 168.2501 | 0.0000 |
| 8.2300 | -156.4357 |

| | C | | D | |
|---|---|---|---|---|
| 0.0148 | -0.0113 | 1.2712 | 0.0171 | 0.1017 |
| 1.8286 | -2.8858 | 0.0000 | 0.0000 | 0.0000 |
| 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 0.1448 | 0.4082 | 0.0000 | 0.0000 | 0.0000 |
| -0.1867 | -0.7801 | 1.8675 | 3.6216 | 0.0000 |
| 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| -0.6082 | 0.7857 | 0.0000 | 3.9577 | 0.0000 |
| 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

131

Table (rotated). Columns: No. | %N | A | B | C | D

**No. 7, %N 76**

A:

| -3.4027 | 7.0954 | -10.9750 |
|---|---|---|
| 152.4944 | -384.9074 | 212.9847 |
| 7.3849 | 113.1657 | -812.8875 |

B:

| 2.0319 | 0.0000 |
|---|---|
| 168.2501 | 0.0000 |
| 8.2300 | -167.0033 |

C:

| 0.0164 | -0.0124 | 1.2323 |
|---|---|---|
| 1.8587 | -2.8451 | 0.0000 |
| 1.0000 | 0.0000 | 0.0000 |
| 0.1373 | 0.3962 | 0.0000 |
| -0.2148 | -0.7745 | 1.8649 |
| 0.0000 | 0.0000 | 1.0200 |
| 0.0000 | 1.0400 | 0.0000 |
| -0.6610 | 0.7950 | 0.0000 |
| 0.6044 | 0.0000 | 0.0000 |

D:

| 0.0179 | 0.1142 |
|---|---|
| 0.0000 | 0.0000 |
| 0.0000 | 0.0000 |
| 3.5425 | 0.0000 |
| 0.0000 | 0.0000 |
| 0.0000 | 0.0000 |
| 3.8743 | 0.0000 |
| 0.0000 | 0.0000 |

**No. 8, %N 77**

A:

| -3.5648 | 7.2133 | -11.0303 |
|---|---|---|
| 162.5057 | -391.1026 | 214.0682 |
| 7.9436 | 113.3870 | -757.7993 |

B:

| 2.0219 | 0.0000 |
|---|---|
| 168.3341 | 0.0000 |
| 8.3191 | -177.8234 |

C:

| 0.0182 | -0.0134 | 1.2056 |
|---|---|---|
| 1.9044 | -2.7956 | 0.0000 |
| 1.0000 | 0.0000 | 0.0000 |
| 0.1294 | 0.3849 | 0.0000 |
| -0.2467 | -0.7686 | 1.8603 |
| 0.0000 | 0.0000 | 1.0200 |
| 0.0000 | 1.0400 | 0.0000 |
| -0.7224 | 0.8013 | 0.0000 |
| 0.6044 | 0.0000 | 0.0000 |

D:

| 0.0187 | 0.1277 |
|---|---|
| 0.0000 | 0.0000 |
| 0.0000 | 0.0000 |
| 3.4632 | 0.0000 |
| 0.0000 | 0.0000 |
| 0.0000 | 0.0000 |
| 3.7907 | 0.0000 |
| 0.0000 | 0.0000 |

**No. 9, %N 78**

A:

| -3.7499 | 6.8617 | -11.0696 |
|---|---|---|
| 169.7130 | -341.1479 | 215.4293 |
| 8.3643 | 116.5880 | -691.0250 |

B:

| 2.0106 | 0.0000 |
|---|---|
| 168.4203 | 0.0000 |
| 8.3737 | -192.1109 |

C:

| 0.0201 | -0.0081 | 1.1616 |
|---|---|---|
| 1.8914 | -1.9938 | 0.0000 |
| 1.0000 | 0.0000 | 0.0000 |
| 0.1398 | 0.2281 | 0.0000 |
| -0.2538 | -1.0327 | 1.8446 |
| 0.0000 | 0.0000 | 1.0200 |
| 0.0000 | 1.0400 | 0.0000 |
| -0.7433 | 0.3197 | 0.0000 |
| 0.6044 | 0.0000 | 0.0000 |

D:

| 0.0198 | 0.1469 |
|---|---|
| 0.0000 | 0.0000 |
| 0.0000 | 0.0000 |
| 3.3713 | 0.0000 |
| 0.0000 | 0.0000 |
| 0.0000 | 0.0000 |
| 3.6928 | 0.0000 |
| 0.0000 | 0.0000 |

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 79 | -3.9304 | 6.7530 | -11.1069 | 1.9977 | 0.0000 | 0.0215 | -0.0090 | 1.1228 | 0.0210 | 0.1695 |
| | | 171.3819 | -346.7343 | 216.9203 | 168.5107 | 0.0000 | 1.4224 | -1.9691 | 0.0000 | 0.0000 | 0.0000 |
| | | 8.5109 | 117.1458 | -630.5989 | 8.4282 | -207.8982 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1735 | 0.2640 | 1.8232 | 3.2727 | 0.0000 |
| | | | | | | | -0.2240 | -0.9714 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 3.5871 | 0.0000 |
| | | | | | | | -0.7030 | 0.3788 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | | |
| 11 | 80 | -4.0688 | 6.6675 | -11.2039 | 2.0073 | 0.0000 | 0.0220 | -0.0097 | 1.1310 | 0.0214 | 0.1777 |
| | | 172.0653 | -350.7980 | 216.2382 | 168.5567 | 0.0000 | 1.2631 | -1.8946 | 0.0000 | 0.0000 | 0.0000 |
| | | 8.7006 | 116.2872 | -619.5431 | 8.5657 | -213.2822 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1987 | 0.2995 | 1.8134 | 3.1781 | 0.0000 |
| | | | | | | | -0.1810 | -0.9148 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 3.4894 | 0.0000 |
| | | | | | | | -0.6505 | 0.4387 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | | |
| 12 | 81 | -4.1079 | 6.6110 | -11.3180 | 2.0172 | 0.0000 | 0.0250 | -0.0105 | 1.1478 | 0.0215 | 0.1827 |
| | | 194.1560 | -355.9573 | 215.3129 | 168.6027 | 0.0000 | 1.4190 | -1.8972 | 0.0000 | 0.0000 | 0.0000 |
| | | 9.9947 | 115.1995 | -618.6472 | 8.7390 | -216.5712 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1047 | 0.3352 | 1.8089 | 3.0909 | 0.0000 |
| | | | | | | | -0.3019 | -0.8596 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 3.3998 | 0.0000 |
| | | | | | | | -0.8539 | 0.5039 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | | |

| No. | %N | A | | | B | | | | C | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | 0.0264 | -0.0111 | 1.1537 | 0.2200 | 0.1919 | |
| | | | | | | | 1.4207 | -1.8641 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 0.0877 | 0.3434 | 0.0000 | 0.0000 | 0.0000 | |
| 13 | 82 | -4.1875 | 6.6018 | -11.4130 | 2.0313 | 0.0000 | -0.3087 | -0.8326 | 1.7947 | 2.9954 | 0.0000 | |
| | | 201.1945 | -358.1946 | 214.3934 | 168.6384 | 0.0000 | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 | |
| | | 10.5542 | 114.2531 | -606.3780 | 8.8545 | -222.0556 | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | -0.8780 | 0.5200 | 0.0000 | 3.3013 | 0.0000 | |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 0.0274 | -0.0116 | 1.1624 | 0.0224 | 0.2007 | |
| | | | | | | | 1.4010 | -1.8302 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 0.0927 | 0.3500 | 0.0000 | 0.0000 | 0.0000 | |
| 14 | 83 | -4.2924 | 6.5964 | -11.5095 | 2.0451 | 0.0000 | -0.2862 | -0.8088 | 1.7814 | 2.9043 | 0.0000 | |
| | | 204.5892 | -360.2404 | 213.4684 | 168.6741 | 0.0000 | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 | |
| | | 10.9348 | 113.3106 | -596.1369 | 8.9705 | -222.0556 | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | -0.8567 | 0.5326 | 0.0000 | 3.2075 | 0.0000 | |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 0.0284 | -0.0121 | 1.1721 | 0.0227 | 0.2099 | |
| | | | | | | | 1.9338 | -1.9056 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | 0.0979 | 0.3565 | 0.0000 | 0.0000 | 0.0000 | |
| 15 | 84 | -4.4119 | 6.5831 | -11.6061 | 2.0565 | 0.0000 | -0.2671 | -0.7872 | 1.7693 | 2.8186 | 0.0000 | |
| | | 208.7422 | -361.9898 | 212.6907 | 168.7151 | 0.0000 | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 | |
| | | 11.3540 | 112.4858 | -586.6067 | 9.0904 | -232.5633 | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 | |
| | | | | | | | -0.8409 | 0.5427 | 0.0000 | 3.1188 | 0.0000 | |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | |

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 85 | -4.5999 | 6.5570 | -11.6598 | 2.0399 | 0.0000 | 0.0305 | -0.0127 | 1.1570 | 0.0235 | 0.2312 |
| | | 216.7323 | -365.4120 | 214.0861 | 168.8145 | 0.0000 | 1.9064 | -1.8457 | 0.0000 | 0.0000 | 0.0000 |
| | | 11.8577 | 113.0807 | -557.0501 | 9.1891 | -245.9792 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1047 | 0.3630 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.2756 | -0.7654 | 1.7536 | 2.7438 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.8620 | 0.5530 | 0.0000 | 3.0373 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 17 | 86 | -4.3035 | 6.5169 | -11.7150 | 2.0245 | 0.0000 | 0.0310 | -0.0131 | 1.1479 | 0.0242 | 0.2524 |
| | | 214.0646 | -368.0331 | 215.3978 | 168.9128 | 0.0000 | 1.8183 | -1.7827 | 0.0000 | 0.0000 | 0.0000 |
| | | 11.7950 | 113.6631 | -533.0996 | 9.2891 | -258.8681 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0483 | 0.3695 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3270 | -0.7454 | 1.7384 | 2.6717 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.9125 | 0.5597 | 0.0000 | 2.9588 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 18 | 87 | -4.5406 | 6.4438 | -11.7599 | 2.0120 | 0.0000 | 0.0335 | -0.0134 | 1.1386 | 0.0249 | 0.2742 |
| | | 224.7873 | -369.0650 | 216.6676 | 169.0048 | 0.0000 | 1.8184 | -1.7124 | 0.0000 | 0.0000 | 0.0000 |
| | | 12.4651 | 114.3035 | -511.5455 | 9.3579 | -271.6347 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0551 | 0.3759 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3379 | -0.7277 | 1.7225 | 2.6026 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.9430 | 0.5598 | 0.0000 | 2.8840 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

135

| No. | %N | A | | | B | | C | | | D | |
|-----|-----|------|------|------|------|------|------|------|------|------|------|
| 19 | 88 | -4.8189 | 6.3078 | -11.8134 | 1.9985 | 0.0000 | 0.0362 | -0.0131 | 1.1343 | 0.0256 | 0.2963 |
| | | 237.5202 | -366.7428 | 217.8890 | 169.1008 | 0.0000 | 1.8278 | -1.6171 | 0.0000 | 0.0000 | 0.0000 |
| | | 13.2514 | 115.1054 | -493.6416 | 9.4496 | -284.2652 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0623 | 0.3824 | 1.7067 | 2.5338 | 0.0000 |
| | | | | | | | -0.3508 | -0.7160 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 2.8092 | 0.0000 |
| | | | | | | | -0.9796 | 0.5447 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 20 | 89 | -5.1256 | 6.1929 | -11.8883 | 1.9904 | 0.0000 | 0.0391 | -0.0128 | 1.1381 | 0.0261 | 0.3153 |
| | | 251.4529 | -364.8737 | 218.4999 | 169.1863 | 0.0000 | 1.5576 | -1.4979 | 0.0000 | 0.0000 | 0.0000 |
| | | 14.1503 | 115.4674 | -481.8437 | 9.5738 | -294.7610 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0695 | 0.3888 | 1.6916 | 2.4633 | 0.0000 |
| | | | | | | | -0.3584 | -0.7025 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 2.7332 | 0.0000 |
| | | | | | | | -1.0130 | 0.5339 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 21 | 90 | -5.4562 | 6.0247 | -11.9894 | 1.9899 | 0.0000 | 0.0420 | -0.0121 | 1.1501 | 0.0265 | 0.3298 |
| | | 266.2453 | -359.3898 | 218.2949 | 169.2554 | 0.0000 | 1.5794 | -1.4138 | 0.0000 | 0.0000 | 0.0000 |
| | | 15.1661 | 115.4849 | -476.0056 | 9.7331 | -302.3803 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0766 | 0.3953 | 1.6778 | 2.3906 | 0.0000 |
| | | | | | | | -0.3575 | -0.6933 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 2.6558 | 0.0000 |
| | | | | | | | -1.0390 | 0.5108 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 22 | 91 | -6.5777 | 5.9865 | -12.1171 | 1.9861 | 0.0000 | 0.0483 | -0.0119 | 1.1670 | 0.0268 | 0.3437 |
| | | 303.5628 | -357.6932 | 217.9458 | 169.3351 | 0.0000 | 1.6916 | -1.3487 | 0.0000 | 0.0000 | 0.0000 |
| | | 17.4855 | 115.2084 | -472.7288 | 9.9605 | -309.7779 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1584 | 0.3901 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3139 | -0.6886 | 1.6640 | 2.3159 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -1.0574 | 0.4897 | 0.0000 | 2.5754 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 23 | 92 | -6.7087 | 6.2445 | -12.2541 | 1.9817 | 0.0000 | 0.0506 | -0.0136 | 1.1872 | 0.0271 | 0.3574 |
| | | 314.4742 | -368.1513 | 217.5057 | 169.4173 | 0.0000 | 1.6814 | -1.3496 | 0.0000 | 0.0000 | 0.0000 |
| | | 18.3398 | 114.1595 | -470.5081 | 10.2082 | -317.0099 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1359 | 0.3724 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3294 | -0.6780 | 1.6501 | 2.2423 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -1.0933 | 0.5029 | 0.0000 | 2.4959 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 24 | 93 | -6.8444 | 6.5214 | -12.3778 | 1.9797 | 0.0000 | 0.0530 | -0.0155 | 1.2046 | 0.0274 | 0.3718 |
| | | 325.9012 | -379.5685 | 217.0736 | 169.4930 | 0.0000 | 1.6755 | -1.3550 | 0.0000 | 0.0000 | 0.0000 |
| | | 19.2424 | 113.0409 | -467.3922 | 10.4172 | -324.3030 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1133 | 0.3547 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3450 | -0.6672 | 1.6357 | 2.1727 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -1.1302 | 0.5185 | 0.0000 | 2.4211 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

137

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 25 | 94 | -7.0767 | 6.5463 | -12.4903 | 1.9796 | 0.0000 | 0.0563 | -0.0154 | 1.2205 | 0.0278 | 0.3872 |
| | | 342.6029 | -378.0984 | 216.6558 | 169.5631 | 0.0000 | 1.9572 | -1.2928 | 0.0000 | 0.0000 | 0.0000 |
| | | 20.4782 | 112.7050 | -463.3129 | 10.5935 | -331.8146 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0904 | 0.3371 | 0.0000 | 2.1059 | 0.0000 |
| | | | | | | | -0.3672 | -0.6740 | 1.6203 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 2.3495 | 0.0000 |
| | | | | | | | -1.1860 | 0.4829 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | | |
| 26 | 95 | -7.4870 | 6.3777 | -12.5656 | 1.9650 | 0.0000 | 0.0593 | -0.0138 | 1.2275 | 0.0284 | 0.4149 |
| | | 353.6834 | -368.3812 | 217.5921 | 169.6689 | 0.0000 | 1.9095 | -1.1972 | 0.0000 | 0.0000 | 0.0000 |
| | | 21.2670 | 113.8256 | -451.5867 | 10.7355 | -345.5432 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1288 | 0.3194 | 0.0000 | 2.0399 | 0.0000 |
| | | | | | | | -0.3358 | -0.6897 | 1.5978 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 2.2767 | 0.0000 |
| | | | | | | | -1.1531 | 0.4127 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | | |
| 27 | 96 | -7.3803 | 6.2548 | -12.6581 | 1.9500 | 0.0000 | 0.0568 | -0.0124 | 1.2378 | 0.0289 | 0.4401 |
| | | 333.6413 | -360.3878 | 218.3313 | 169.7787 | 0.0000 | 1.6854 | -1.0899 | 0.0000 | 0.0000 | 0.0000 |
| | | 20.1890 | 114.7311 | -443.7770 | 10.9162 | -357.9977 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1842 | 0.3012 | 0.0000 | 1.9780 | 0.0000 |
| | | | | | | | -0.2466 | -0.7051 | 1.5790 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 2.2081 | 0.0000 |
| | | | | | | | -0.9842 | 0.3500 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | | |

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 28 | 97 | -7.4061 | 6.3277 | -12.7448 | 1.9375 | 0.0000 | 0.0590 | -0.0126 | 1.2512 | 0.0293 | 0.4647 |
| | | 342.0948 | -362.1939 | 218.9814 | 169.8827 | 0.0000 | 1.6548 | -1.0429 | 0.0000 | 0.0000 | 0.0000 |
| | | 20.8304 | 114.9677 | -436.9613 | 11.0762 | -369.8726 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1550 | 0.2829 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.2761 | -0.7086 | 1.5613 | 1.9200 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -1.0269 | 0.3246 | 0.0000 | 2.1443 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 29 | 98 | -7.4573 | 6.4690 | -12.8170 | 1.9275 | 0.0000 | 0.0617 | -0.0135 | 1.2606 | 0.0298 | 0.4909 |
| | | 352.2945 | -367.5496 | 219.6499 | 169.9794 | 0.0000 | 1.6352 | -1.0165 | 0.0000 | 0.0000 | 0.0000 |
| | | 21.5905 | 114.9815 | -429.5205 | 11.1969 | -382.0276 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1253 | 0.2646 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3076 | -0.7073 | 1.5424 | 1.8641 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -1.0754 | 0.3124 | 0.0000 | 2.0830 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 30 | 99 | -7.6395 | 6.5649 | -12.8752 | 1.9197 | 0.0000 | 0.0654 | -0.0139 | 0.5452 | 0.0301 | 0.5158 |
| | | 369.7002 | -370.6537 | 220.3537 | 170.0697 | 0.0000 | 1.6494 | -0.9809 | 0.0000 | 0.0000 | 0.0000 |
| | | 22.8114 | 115.1478 | -424.5239 | 11.2804 | -394.5778 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.0949 | 0.2463 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -0.3477 | -0.7081 | 1.5220 | 1.8099 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | -1.1482 | 0.2925 | 0.0000 | 2.0239 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

| No. | %N | A | | | B | | C | | | D | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 31 | 100 | -3.2992 | 6.4171 | -12.9402 | 1.9188 | 0.0000 | 0.0149 | -0.0120 | 0.5509 | 0.0299 | 0.5282 |
| | | 85.2022 | -360.8143 | 220.5113 | 170.1407 | 0.0000 | 0.2109 | -0.8924 | 0.0000 | 0.0000 | 0.0000 |
| | | 5.1168 | 115.7660 | -429.1746 | 11.3543 | -405.7120 | 1.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.3263 | 0.2283 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.2141 | -0.7210 | 1.5006 | 1.7544 | 0.0000 |
| | | | | | | | 0.0000 | 0.0000 | 1.0200 | 0.0000 | 0.0000 |
| | | | | | | | 0.0000 | 1.0400 | 0.0000 | 0.0000 | 0.0000 |
| | | | | | | | 0.1293 | 0.2313 | 0.0000 | 1.9642 | 0.0000 |
| | | | | | | | 0.6044 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |

# NOMENCLATURE

A      Coefficient System Matrix

A      Area ($ft^2$)

$A_{ij}(\omega)$      Lower Bound of System Specification

B      Input System Matrix

$B_{ij}(\omega)$      Upper Bound of System Specification

C      Output System Matrix

C      Controller, Feedback Regulator

C      Proportionality Constant

D      Transmission System Matrix

DoD      Department of Defense

F      Prefilter Matrix

Fn      Thrust

G      Diagonal Controller Matrix

G      Proportional Gain Matrix for Proportional-Plus-Integral Observer

IEC      Intelligent Engine Control

IHPTET      Integrated High Performance Turbine Engine Technology

IMC      Internal Model Control

INA      Inverse Nyquist Array

J      Moment of Inertia

| K | Precompensating Matrix |
|---|---|
| K | Proportionality Constant |
| K | Integral Gain Matrix for Proportional-Plus-Integral Observer |
| M | Reference Model |
| MBC | Model-Based Control |
| MRAC | Model-Reference Adaptive Control |
| N | Speed (rpm) |
| $\dot{N}$ | Acceleration (rpm/sec) |
| NASA | National Aeronautics and Space Administration |
| P | Plant Matrix |
| P | Pressure (lb/in$^2$) |
| PM | Plant Model |
| $\hat{P}$ | Inverse of the Matrix P |
| $P$ | Family of Plants, Plant Set |
| Q | Heating Value of Fuel |
| QFT | Quantitative Feedback Theory |
| QNA | Quantitative Nyquist Array |
| R | Scaling Matrix |
| S | Diagonal Sensitivity Matrix |
| SAE | Society of Automotive Engineers |
| SM | Stall Margin |
| T | Temperature |
| T | Diagonal Complimentary Sensitivity Function |

| | |
|---|---|
| T | Transfer Function Matrix |
| V | Velocity |
| V | Volume |
| $c_p$ | Specific Heat at Constant Pressure |
| e | Error |
| h | Enthalpy |
| k | Ratio of Specific Heats |
| $\ell_i$ | Loop Transmission Element for the ith Loop |
| $\dot{m}$ | Mass flow ($lb_m$/sec) |
| $q_{ij}$ | ith row, jth column of $\hat{P}^{-1}$ |
| s | Laplace Operator |
| u | Input Vector |
| w | Flow |
| x | State Variable Vector |
| $\dot{x}$ | First Derivative of State Variable Vector |
| $\hat{x}$ | Estimated State Variables |
| y | Output Vector |
| $\alpha$ | Vector of Parameters Corresponding to Points of Linearization |
| $\gamma$ | Interaction Index |
| $\eta$ | Efficiency |
| $\lambda$ | Eigenvalue |
| $\lambda_{max}$ | Maximum Eigenvalue |

$\lambda_p$     Perron Root

$\rho$     Spectral Radius

$\tau$     Torque

$\omega$     frequency (rad/sec)

## Subscripts

b     Burner, Combustor

c     Compressor

d     Diagonal

f     Fuel

i     ith element

ij     ith input, jth output

o     Off Diagonal

t     Turbine

Y/R     Output to Reference

0 – 9     Engine Station Designations